(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 017 178 A1

(12) EUROPEAN PATENT APPLICATION
published in accordance with Art. 158(3) EPC

(43) Date of publication:
05.07.2000 Bulletin 2000/27

(21) Application number: 99919618.1

(22) Date of filing: 17.05.1999

(51) Int. Cl.7: H03M 13/23

(86) International application number:
PCT/JP99/02553

(87) International publication number:
WO 99/62183 (02.12.1999 Gazette 1999/48)

(84) Designated Contracting States:
DE ES FR GB IT

(30) Priority: 28.05.1998 JP 14676298
25.08.1998 JP 23898798

(71) Applicant: Sony Corporation
Tokyo 141-0001 (JP)

(72) Inventors:
- MIYAUCHI, Toshiyuki
Sony Corporation
Tokyo 141-0001 (JP)
- HATTORI, Masayuki
Sony Corporation
Tokyo 141-0001 (JP)

(74) Representative:
Melzer, Wolfgang, Dipl.-Ing. et al
Patentanwälte
Mitscherlich & Partner,
Sonnenstrasse 33
80331 München (DE)

(54) SOFT OUTPUT DECODER FOR CONVOLUTION CODE AND SOFT OUTPUT DECODING METHOD

(57) A soft output decoding method and apparatus for convolutional codes. After computing the number of states times Iβ ($\beta_t \sim \beta_{t-D+1}$) for a truncated length, the soft output outside the truncated length is sequentially computed, as next following Iβ ($\beta_{t-D} \sim \beta_{t-2D+1}$) outside the next following truncated length is computed, at the same time as Iβ of the next truncated length is sequentially computed. In this manner, a decoder 4 performs computation of Iβ within the truncated length and computation of Iβ retrogressive by not less than the truncated length, as parallel processing, as a result of which the computation of Iβ per clock is the number of states ×2 to reduce the volume of computation to expedite the decoding.

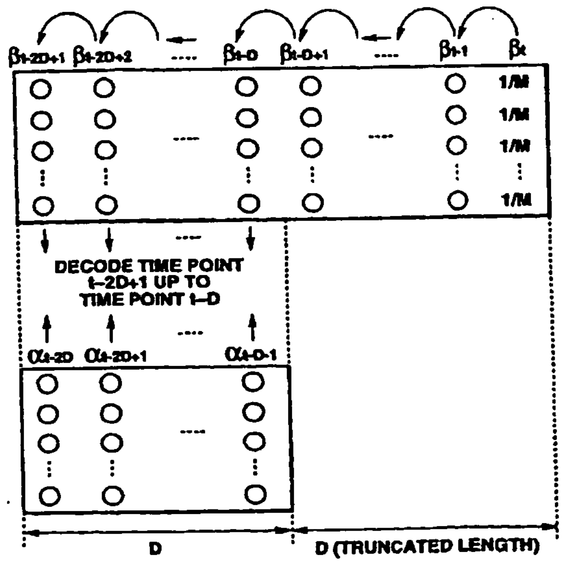


FIG.7

EP 1 017 178 A1

Description

Technical Field

[0001] This invention relates to a soft output decoding method and apparatus for convolutional codes, applied with advantage to, for example, a satellite broadcast reception device. More particularly, it relates to a soft output decoding device in which the probability information is stored on a recording medium a length not less than a truncated length and in which updating of the probability information within the truncation length and computations of the soft output outside the truncated length are executed in parallel to enable high-speed operations.

Background Art

[0002] As a decoding method for minimizing the symbol error ratio following decoding of the convolutional codes, there is known a BCJR algorithm from Bahl, Cocke, Jelinek and Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate", IEEE Trans. Information. Theory, Vol.1T-20, pp. 284 to 287, Mar. 1974. In the BCJR algorithm, it is not each symbol, but the likelihood of each symbol, that is outputted as the decoding result. This output is termed soft output.

[0003] In these years, researches are being conducted towards reducing the symbol error rate by using soft outputs as decoded output of the inner code of the conjugated code or as outputs of each reiteration of the iterative decoding method. As a decoding method, suited to this purpose, a BCJR algorithm is stirring up notice.

[0004] The contents of the BCJR algorithm are hereinafter explained in detail.

[0005] The BCJR algorithm outputs the likelihood of each symbol, instead of outputting each symbol, as a decoding result. The BCJR algorithm is used when convolutional encoding the digital information is convolutional-encoded by a convolutional encoder and the resulting data string obtained on convolutional encoding is observed over a non-storage communication route.

[0006] M states (transition states) representing the contents of shift registers of a convolutional encoder are expressed as m(0, 1, • • •, M|1). The state at time t is S_t, an input at t is I_t, an output at time t is X_t, and an output sequence is $X_t^{t'} = X_t, X_{t+1}, \cdots, X_{t'}$. The transition probability between respective states $p_t(m|m')$ is represented by the following equation (1):

$$p_t(m|m') = \Pr\{S_t = m|S_{t-1} = m'\} \quad (1).$$

[0007] It is noted tat Pr{A|B} is a conditional probability of occurrence of A, under a condition that an event B has occurred, while Pr{A;B} is a probability of occurrence of both A and B. It is also assumed that the convolutional codes by the convolutional encoder starts with the state $S_0 = 0$ and outputs X_1^{τ} to terminate at $S_{\tau} = 0$.

[0008] The noisy non-storage communication route receives an output X_1^{τ} as an input and outputs Y_1^{τ}. It is assumed that an output sequence $Yt^{t'} = Y_t, Y_{t+1}, \cdots, Y_{t'}$. Meanwhile, the transition probability of the non-storage communication route can be defined, for all t ($1 \le t \le \tau$), by a function R(• | •) satisfying the following equation (2):

$$\Pr\{Y_1{}^t|X_1{}^t\} = \prod_{j=1}^{t} R(Y_j|X_j) \quad (2).$$

[0009] Therefore, if the probability λ_t is defined as in the following equation (3):

$$\Lambda t = \frac{\{\Pr I_t = 1|Y_1{}^{\tau}\}}{\{\Pr I_t = 0|Y_1{}^{\tau}\}} \quad (3)$$

this probability λ_t represents the likelihood of the input information at time t when Y_1^{τ} is received, such that this probability λ_t is the soft output to be found.

[0010] The BCJR algorithm defines the probabilities α_t, β_t and γ_t as indicated by the following equations (4) to (6):

$$\alpha_t(m) = \Pr\{S_t = m; Y_1{}^t\} \quad (4)$$

$$\beta_t(m) = \Pr\{T_{t+1}{}^{\tau}|S_t = m\} \quad (5)$$

$$\gamma_t(m', m, i) = \Pr\{St = m; Y_t; i_t = i | S_{t-1} = m'\} \quad (6).$$

[0011] The contents of α_t, β_t and γ_t are explained with reference to Fig. 1 which illustrates the relation between the respective probabilities. It is noted that α_{t-1} corresponds to the probability of passing through respective states at time t-1, as computed based on a reception word as from the encoding starting state S0 = 0, while β_t corresponds to the probability of passing through respective states at time T as computed in the reverse sequence to the chronological sequence based on the reception word as from the encoding end state Sτ = 0, and γ_t corresponds to the reception probability of respective branches in transition through respective states at time t as computed based on the reception word and the input probability at time t.

[0012] With the aid of α_t, β_t and γ_t, the soft output λ_t can be represented by the following equation (7):

$$\Upsilon_t = \frac{\sum_{m=0}^{M-1}\sum_{m'=0}^{M-1} \alpha_{t-1}(m')\chi_t(m',m,1)\beta_t(m)}{\sum_{m=0}^{M-1}\sum_{m'=0}^{M-1} \alpha_{t-1}(m')\chi_t(m',m,0)\beta_t(m)} \quad \cdots\cdots (7).$$

[0013] It is noted that, for t = 1, 2, • • •, τ, the following equation (8) holds:

$$\alpha_t(m) = \sum_{m=0}^{M-1}\sum_{t'=0}^{1} \alpha_{t-1}(m')\chi_t(m',m,i) \quad (8).$$

where $\alpha_0(0) = 1$, $\alpha_0(m) = 0$ $(M \neq 0)$.

[0014] Similarly, the following equation (9):

$$\beta_t(m) = \sum_{m=0}^{M-1}\sum_{i=0}^{1} \beta_{t+1}(m')\chi_{t+1}(m',m,i) \quad (9)$$

holds for t = 1, 2, • • •, τ - 1, where $\beta_\gamma(0) = 1$, $\beta_\tau(m) = 0$ $(m \neq 0)$.

[0015] For γ_t, the following equation (10) holds:

$\gamma_t(m',m,i) = P_t(m|m')R(Y_t, X)$ in case of transition from m' to m for an input i (X being its output); and (10).

$\gamma_t(m',m,1) = 0$ in case of non-transition from m' to m for the input i

[0016] Based on the above equation, the BCJR algorithm finds the soft output λ_t in accordance with the following sequence (a) to (c):

(a) each time Y_t is received, $\alpha_t(m)$, $\gamma_t(m',m,i)$ is computed using the equations (8) and (10);
(b) after reception of the entire sequence Y_1^τ, $\beta_t(m)$ is computed for respective states m for the totality of time points t, using the equation (9); and
(c) α_t, β_t and γ_t, computed at (a) and (b), are substituted into the equation (7) to compute the soft output λ_t at each time point t.

[0017] Meanwhile, the above-described BCJR algorithm suffers from the problem that the volume of computation is considerable because of the product computations involved and that continuous data cannot be received because of the necessity of terminating the codes.

[0018] In order to combat these problems, a Max-Log-BCJR algorithm and a log-BCJR algorithm have been proposed as techniques for diminishing the computation volume in Robertson, Villebrun and Hoeher, "A Comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in the Domain", in IEEE Int. Cof. On Communications, pp. 1009 to 1013, June 1995,while the SW-BCJR algorithm for performing sliding window processing has been proposed as a technique for receiving continuous data in Benedetto and Montorsi, "Soft-Output Decoding Algorithm in Iterative Decoding of Turbo Codes", TDA Progress Report 42-124, Feb. 1996.

[0019] The contents of these algorithms are hereinafter explained.

[0020] First, the contents of the Max-Log-BCJR algorithm and the log-BCJR algorithm are explained.

[0021] The Max-Log-BCJR algorithm represents the probabilities α_t, β_t and γ_t, λ_t by a modulo 2 logarithm (natural logarithm) and substitutes the logarithmic sum computations for the product processing of the probabilities, as shown in the following equation (11), while approximating the logarithmic sum computations by the logarithmic maximum value computations, as shown in the following equation (12). Meanwhile, max(x, y) is a function which selects a larger one of x or y.

$$\mathrm{Log}(e^{x} \cdot e^{y}) = x + y \tag{11}$$

$$\mathrm{Log}(e^{x} + e^{y}) = \max(x, y) \tag{12}$$

[0022] For simplifying the explanation, logarithms of α_t, β_t and γ_t, λ_t are set as $I\alpha_t$, $I\beta_t$, $I\gamma_t$, $I\lambda_t$, respectively, as shown by the following equations (13) to (15):

$$I\alpha_t(m) = \log(\alpha_t(m)) \tag{13}$$

$$I\beta_t(m) = \log(\beta_t(m)) \tag{14}$$

$$I\gamma_t(m) = \log(\gamma_t(m)) \tag{15}$$

where "I" denotes the modulo e logarithm.

[0023] In the Max-Log-BCJR algorithm, these $I\alpha_t$, $I\beta_t$, $I\gamma_t$ are approximated as shown by the following equations (16) to (18):

$$I\alpha_t m = \max_{i=0,1} \max_{m'} (I\alpha_{t-1}(m') + I\chi_t(m', m, i)) \quad \cdots(16)$$

$$I\beta_t(m) = \max_{i=0,1} \max_{m'} (I\beta_{t+1}(m') + I\chi_t(m', m, i)) \quad \cdots(17)$$

$$I\gamma_t(m', m, i) = \log(P_t(m|m')) + \log(R(Y_t, X)) \tag{18}$$

where X is an encoder output on transition from m' to m. It is noted that max m' in $I\alpha_t(m)$ and $I\beta_t(m)$ is to be found in the state m' in which transition to a state m occurs for the input i.

[0024] Similarly, $I\lambda_t$ is approximated as shown in the following equation (19), in which max m' in the first term of the right side is to be found in the state m' where transition to the state m exists for the input = 1 and in which max m' in the second term is to be found in the state m' where transition to the state m exists for the input = 0.

$$I\lambda_t = \max_{i=0.1} \max_{m'} (I\alpha_{t-1}(m') + I\chi_t(m',m,1) + I\beta_t(m))$$
$$- \max_{m=0,\ldots,m=1} \max_{m'} (I\alpha_{t-1}(m') + I\chi_t(m',m,0) + I\beta_t(m))$$

$\cdots(19).$

[0025] Based on the above relation, the Max-Log-BCJR algorithm finds the soft output λ_t in accordance with the following sequence (a) to (c):

(a) each time Y_t is received, $I\alpha_t(m)$ and $I\gamma_t(m',m,1)$ are computed, using the equations (16) and (18);
(b) after receiving the sequence Y_1^τ in its entirety, $I\beta_t(m)$ is computed, for all states m of all time points t, using the equation (17);
(c) $I\alpha_t$, $I\beta_t$ and $I\gamma_t$, computed in (a) to (c), are substituted into the equation (19) to compute the soft output $I\lambda_t$ at each time point.

[0026] Since there are included no product computations in the Max-Log-BCJR algorithm, the processing volume can be reduced significantly in comparison with the BCJR algorithm.
[0027] Meanwhile, if the probability sum computation is modified as shown in the following equation (20):

$$\mathrm{Log}(e^x + e^y) = \max(x, y) + \log(1 + e^{-|X-Y|} \qquad (20)$$

the second term of the right side becomes the linear function for the variable |x - y|, so that, by corresponding tabulation, the logarithmic values of the sum processing can be found correctly.
[0028] The Log-BCJR algorithm substitutes the equation (20) for the equation (12) in the Max-Log-BCJR algorithm in its entirety to realize correct probability computations. As compared to the Max-Log-BCJR algorithm, the Log-BCJR algorithm is increased in processing volume. However, there is included no product computation in the Log-BCJR algorithm, such that its output is no other than the logarithmic value of the soft output of the BCJR algorithm except the quantization error.
[0029] Since the second term of the right side in the equation (20) is the linear function by the variable |x - y|, it is possible to obtain simple and highly accurate results of computation by, for example, tabulation. Thus, a soft output higher in accuracy can be obtained with the Log-BCJR algorithm than is possible with the Max-Log-BCJR algorithm.
[0030] The contents of the SW-BCJR algorithm are hereinafter explained.
[0031] In the BCJR algorithm, the code needs to be terminated for γ_t computation, such that continuous data cannot be received. The BCJR algorithm accords 1/M for the totality of states as an initial value of β_t, introduces the truncated length as in the case of Viterbi decoding and finds the soft output by retrograding down the time axis by a truncated length D as set (see Fig.2).
[0032] The SW-BCJR algorithm initializes α_0, as in the case of the usual BCJR algorithm, and performs the operations (a) to (e) for each time instant to find the soft output for each time instant.

(a) γ_t is found based on the received value at time t and the transition probability;

(b) $\beta_t(m)$ is initialized for the totality of states m so that $\beta_t(m) = 1/M$;

(c) $\beta_{t-1}, \cdots, \beta_{t-D}$ are computed based on $\gamma_{t-D-1}, \cdots, \gamma_t$;

(d) from β_t-D and α_{t-D-1}, as found, the soft output γ_{t-D} at time t-D is found by the following equation (21);

$$\gamma_{t-D} \quad \frac{\Sigma\alpha_{t-D-1}(m',m,1) + I\beta_{t-D}(m)}{\Sigma\alpha_{t-D-1}(m',m,0) + I\beta_{t-D}(m)} \qquad (21)$$

and

(e) from α_{t-D-1} and γ_{t-D}, α_{t-D} is computed.

[0033] In the above-given thesis by Benedetto et al., there are also proposed a SW-Log-BCJR algorithm, combined from the SW-BCJR algorithm and the Log-BCJR algorithm, and a SW-Max-Log-BCJR algorithm, combined from the SW-BCJR algorithm and the Max-Log-BCJR algorithm. It is noted that the SW-Max-Log-BCJR algorithm is referred to in the thesis as SWAL-BCJR algorithm.

[0034] With use of the SW-Max-Log-BCJR algorithm or the SW-Log-BCJR algorithm, it is possible to receive continuous data to find a soft output. However, in these algorithms, in contradistinction from the case of decoding terminated codes, it is necessary to find the number of states multiplied by Y_t for the truncated length in order to find one decoded output, with the result that a large processing volume is involved for mounting even though the product computations are nit involved.

[0035] The SW-Max-Log-BCJR algorithm or the SW-Log-BCJR algorithm has a drawback that, while it is possible to receive the convolutional encoded and transmitted continuous data to find the soft output, the processing volume for producing a code output is increased to render high-speed processing difficult.

[0036] On the other hand, with the SW-Log-BCJR algorithm, combined from the SW-BCJR algorithm and the Log-BCJR algorithm, or the SW-Max-Log-BCJR algorithm, combined from the SW-BCJR algorithm and the Max-Log-BCJR algorithm, also termed the SWAL-MAP algorithm, it is possible to diminish the processing volume to find the soft output of continuous data.

[0037] However, with these algorithms, it is necessary to retrograde down the time axis by the truncated length D to produce a soft output to find β for the number of states times the truncated length D, with the result that a tremendous processing volume is required despite the fact that product processing is not involved in the processing.

Disclosure of the Invention

[0038] It is therefore an object of the present invention to provide a method and apparatus for decoding a soft output of convolutional codes with a high-speed operation.

[0039] It is another object of the present invention to provide a method and apparatus for decoding a soft output of convolutional codes by a simplified structure.

[0040] According to the present invention, when finding the probability information at each transition state of the convolutional code, and computing and outputting a soft output using the probability information, the probability information is partitioned in terms of a pre-set truncated length as a unit and stored. The updating of the probability information in the truncated length and the computation of the soft output outside the truncated length are executed in parallel.

[0041] According to the present invention, a smaller processing volume per clock or a smaller amount of accessing to the memory suffices to enable a high-speed processing.

[0042] Thus, in one aspect of the present invention, there is provided a soft output decoding apparatus for convolutional codes probability computing means for finding the probability information in each transition state of the convolutional codes, probability storage means for storing the probability information as found by the probability computing means in a recording medium, and soft output computing means for finding a soft output using the probability information stored in the recording medium. The probability storage means stores the probability information in an amount not less than a truncated length on the recording medium, and the updating of the probability information within the truncated length by the probability storage means and the computation of the soft output outside the truncated length by the soft output computing means are carried out in parallel.

[0043] In another aspect of the present invention, there is provided a soft output decoding method for convolutional codes including a first step of finding the probability information at each transition state of the convolutional codes, a second step of storing the probability information as found in the first step in the recording medium in the first step a length not less than a truncated length, and a third step of finding a soft output using the probability information stored in the recording medium in the second step. The updating of the probability information within the truncated length in the second step and the computation of the soft output outside the truncated length in the third step are carried out in parallel.

Brief Description of the Drawings

[0044]

Fig.1 illustrates the contents of α_t, β_t and γ_t in the BCJR algorithm.

Fig.2 illustrates the contents of the SW-BCJR algorithm.

Fig.3 is a block diagram showing a communication model embodying the present invention.

Fig.4 is a block diagram showing the structure of a convolutional encoder in the communication model.

Fig.5 shows the trellis of the convolutional encoder.

Fig.6 is a block diagram showing the structure of a decoder in the communication model.

Fig.7 illustrates the sequence of soft output computation in the communication model.

Fig.8 is a block diagram showing the structure of an Iγ computation storage circuit in the decoder shown in Fig.6.

Fig.9 is a timing chart for illustrating the operation of a RAM constituting the Iγ computation storage circuit.

Fig.10 is a block diagram showing the structure of the Iα computation storage circuit in the decoder shown in Fig.6.

Fig.11 is a block diagram showing the structure of an Iα computation circuit in the Iα computation storage circuit.

Fig.12 is a block diagram showing the structure of an addition comparison selection circuit within the Iα computation circuit.

Fig.13 is a timing chart for illustrating the operation of a register and a RAM making up the Iα computation storage circuit.

Fig.14 is a bd showing the structure of the Iβ computation storage circuit.

Fig.15 is a block diagram showing the structure of an Iβ computing circuit in the Iβ computation storage circuit.

Fig.16 is a block diagram for illustrating the structure of an addition comparison selection circuit in the Iβ computation circuit.

Fig.17 is a timing chart for illustrating the operation of the register etc making up the Iβ computation storage circuit.

Fig.18 is a block diagram showing the structure of a soft output computation circuit in the decoder.

Fig.19 is a block diagram showing the structure of an Iλ_1 computation circuit in the soft output computation circuit.

Fig.20 is a block diagram showing die structure of an Iλ_0 computation circuit in the soft output computation circuit.

Fig.21 is a timing chart for illustrating the operation of the soft output computation circuit.

Figs.22A to D illustrate the contents of a memory management in the Iγ computation circuit.

Fig.23 is a timing chart of the memory management.

Fig.24 is a block diagram showing the structure of an addition comparison selection circuit associated with the SW-Log-BCJR algorithm.

Fig.25 is a block diagram showing an alternative structure of the Iγ computation storage circuit in the decoder.

Fig.26 is a timing chart for illustrating the operation of the Iγ computation storage circuit.

Fig.27 is a block diagram showing an alternative structure of an Iβ computation storage circuit in the decoder.

Fig.28 is a timing chart for illustrating the operation of the Iβ computation storage circuit.

Best Mode For Carrying Out the Invention

[0045] Referring to the drawings, preferred embodiments of the present invention will be explained in detail.

[0046] The present invention is applied to a communication model 1 configured as shown for example in Fig.3. This communication model 1 convolutionally encodes the digital information D0 by a convolutional encoder 2 to send the convolutionally encoded data string to a decoder 4 over a noisy non-storage communication route 3 to decode the soft output of the convolutionally encoded data string.

[0047] The convolutional encoder 2 in this communication model 1 is a 1:2 encoder for outputting 2 bits for a 1-bit input. Referring to Fig.4, the convolutional encoder 2 is made up of an input terminal 21, fed with a 1-bit input i_t, output terminals 22a, 22b outputting a 2-bit output X_t, three exclusive-OR circuits (EX • OR circuits) 23 to 25 and two registers 26, 27.

[0048] The input terminal 21 is connected to the output terminal 22a and to an input side of the EX • OR circuit 23, an output of which is connected to the input side of the resistor 26 and to the input side of the EX • OR circuit 24. An output side of the EX • OR circuit 24 is connected to the output terminal 22b, while an output side of the resistor 26 is connected to an input side of the resistor 27 and to an input side of the EX • OR circuit 25. An output of the resistor 27 is connected to the input of the EX • OR circuit 24 and to an input of the EX • OR circuit 25, an output side of which is connected to the input side of the EX • OR circuit 23.

[0049] In this convolutional encoder 2, the 1-bit input i_t, sent to the input terminal 21, is directly outputted at the output terminal 22a, while being inputted to the EX • OR circuit 23. The EX • OR circuit 23 sends an exclusive-OR output between the input i_t and the output of the EX • OR circuit 24 via register 26 and registers 26, 27 to the EX • OR circuit 25. An exclusive-OR output of the EX • OR circuit 24 is fed back to the EX • OR circuit 23. This EX • OR circuit 23 also routes the exclusive-OR output between the input i_t and the output of the EX • OR circuit 25 directly and also via the resistors 26, 27 to the EX • OR circuit 25. The EX • OR circuit 25 outputs an exclusive-OR output between the exclusive-OR output of the EX • OR circuit 23 and an output of the resistor 27 as another bit at the output terminal 22b.

[0050] In the above-described configuration of the convolutional encoder 2, if the 1-bit input i_t is routed to the input terminal 21, a 2-bit output sequence X_t is outputted at the output terminals 22a, 22b. Fig.5 shows the trellis of the convolutional encoder 2, with the number of states being 4.

[0051] The decoder 4 in this communication model 1 is a decoder which is based on the SW-Max-Log-BCJR algorithm and which is associated with the convolutional encoder 2 having a constraint length of 3 shown in Fig.4.

[0052] This decoder 4 processes a received value Y_t of the encoder 2, received from the non-storage communication route 3, with a truncated length D = 4, to output a soft output λ_t. Referring to Fig.6, the decoder 4 includes a controller 41 for controlling the entire operation, input terminals 42y, 43p1 and 42p2, fed with the received value Y_t, a priori probability value $Pr_1 = \log Pr\{i_t = 0\}$ and $Pr_2 = \log Pr\{i_t = 1\}$, respectively, an I$\gamma$ computing circuit 43, an Iα computation storage circuit 44, an Iβ computation storage circuit 45, a soft output computation circuit 46 and an output terminal 47 for outputting a soft output Iλ_t.

[0053] In contradistinction from the usual SW-Max-Log-BCJR algorithm, this decoder 4 does not compute Iβ_t for the number of states x truncated length for decoding for one time juncture. That is, the decoder 4 computes Iβ (shown by $\beta_t \sim \beta_{t-D+1}$), then sequentially computes soft output outside the truncated length, as it computes Iβ outside the truncated length (shown by $\beta_{t-D} \sim \beta_{t-2D+1}$), and sequentially computes Iβ for the next truncated length. Thus, the decoder 4 performs, in parallel, the computation of Iβ in the truncated length and Iβ retrogressive down the time axis by not less than the truncated length, with the computation of Iβ per clock being the number of states x 2.

[0054] The Iγ computation storage circuit 43 is fed from the controller 41 with a control signal Scγ and with the received value Y_t, a priori probability values Pr_1, Pr_2, from the input terminals 42y, 43p1 and 42p2, respectively. The Iγ computation storage circuit 43 uses the received value Y_t, and a priori probability values Pr_1, Pr_2 to compute and store Iγ in accordance with the equation (18) every received value Y_t. The Iγ computation storage circuit 43 then routes Iγ to the computation storage circuit 44, Iβ computation storage circuit 45 and to the soft output computation circuit 46 in a sequence proper for respective processing operations.

[0055] The Iγ computation storage circuit 43 operates as first computation means for computing the first probability γ as determined by the code pattern and by the received value. It is noted that Iγ sent from the Iγ computation storage circuit 43 to the Iα computation storage circuit 44 is depicted as I$\gamma(\alpha)$, Iγ sent from the Iγ computation storage circuit 43 to the Iβ computation storage circuit 45 is depicted as I$\gamma(\beta_1)$, I$\gamma(\beta_2)$ whilst Iγ sent from the Iγ computation storage circuit 43 to the soft output computation circuit 46 is depicted as I$\gamma(\lambda)$.

[0056] The Iα computation storage circuit 44 is fed from the controller 41 with a control signal Scα, while being fed from the Iγ computation storage circuit 43 with I$\gamma(\alpha)$. This Iα computation storage circuit 44 computes and stores Iα, in accordance with the equation (16), using I$\gamma(\alpha)$, to route this I$\gamma(\alpha)$ to the soft output computation circuit 46 in a sequence proper to the processing. The Iα computation storage circuit 44 operates as second computing means for computing the second probability Iα from the encoding starting state chronologically to each state from one received value Y_t to another. Meanwhile, Iα sent from the Iα computation storage circuit 44 to the soft output computation circuit 46 is depicted as Iγ(λ), while Iγ sent to the Iα computation storage circuit 44 is depicted as Iγ(α).

[0057] The Iβ computation storage circuit 45 is fed from the controller 41 with the control signal Scβ, while being fed from the Iγ computation storage circuit 43 with Iγ(β_1) and Iγ(β_2). There is a time shift of the truncation length × 2 between Iγ(β_1) and Iγ(β_2). The Iβ computation storage circuit 45 uses Iγ(β_1) and Iγ(β_2) to compute and store two sequences of Iβ in parallel, in accordance with the equation (17), to send one of the sequences of Iβ to the soft output computation circuit 46 in an order proper to the processing. This Iβ computation storage circuit 45 constitutes third computing means for computing the third probability Iβ from the truncated state to each state in a reverse sequence to the chronological sequence from one received value Yt to another. Meanwhile, Iβ sent from the Iβ computation storage circuit 45 to the soft output computation circuit 46 is termed Iβ(γ).

[0058] The soft output computation circuit 46 is fed from the Iγ computation storage circuit 43 with Iγ(λ), while being fed from the Iα computation storage circuit 44 and the Iβ computation storage circuit 45 with Iα(λ) and Iβ(λ), respectively. The soft output computation circuit 46 uses Iγ(λ), Iα(λ) and Iβ(λ) to compute Iλt in accordance with the equation (19) to chronologically re-array and output the Iγ(λ), Iα(λ) and Iβ(λ).

[0059] The specified structures of the Iγ computation storage circuit 44, Iα computation storage circuit 44, Iβ computation storage circuit 45 and the soft output computation circuit 46 are now explained.

[0060] Fig.8 shows the structure of the Iγ computation storage circuit 43. This Iγ computation storage circuit 43 includes input terminals 301Y, 301P_1, 301P_2 and 301S, fed with the received value Y_t, a priori probability values Pr_1, Pr_2 and with the control signal Scγ, respectively, and read-only memories (ROMs) 302a to 302d constituting a table for outputting the probability IR(Y_t|00), IR(Y_t|01), IR(Y_t|10), IR(Y_t|11) of the received value Y_t for respective states m, with the received value Y_t to the input terminal 301Y as a readout address signal.

[0061] The Iγ computation storage circuit 43 includes adders 303a, 303b for summing the a priori probability information Pr1 supplied to the input terminal 301P_1 for the probability IR(Y_t|00), IR(Y_t|01) outputted by the ROMs 302a, 302b to obtain the probability Iγ[00], Iγ[01] of respective branches associated with outputs [00], [01] on the trellis, and adders 303c, 303d for summing the a priori probability information Pr1 supplied to the input terminal 301P_2 for the probability IR(Y_t|10), IR(Y_t|11) outputted by the ROMs 302c, 302d to obtain the probability Iγ[10], Iγ[11] of respective branches associated with outputs Iγ[10], [11] on the trellis.

[0062] The total number of bits of the respective probabilities Iγ[00] to Iγ[11] outputted by the adders 303a to 303d is the number of bits × 2n in the case of the systematic convolutional code with the code rate of k/n. Thus, the Iγ computation storage circuit 43 sets the respective probabilities Iγ[00] to Iγ[11] with 4 bits and outputs the probabilities Iγ[00] to Iγ[11] wit a sum total of 16 bits.

[0063] The Iγ computation storage circuit 43 also includes random access memories (RAMs) 304a to 304d, adapted for sequentially storing the respective probabilities Iγ[00] to Iγ[11], outputted by the adders 303a to 303d, in accordance with the control signal Scγ and for outputting the respective probabilities in a pre-set sequence, a selection circuit 308 for selectively retrieving Iγ outputted by these RAMs 304a to 304d in accordance with the control signal γ for conversion to Iγ(α) Iγ(β_1), Iγ(β_2) and Iγ(λ) and output terminals 309a to 309d for outputting these Iγ(α) Iγ(β_1), Iγ(β_2) and Iγ(λ).

[0064] In the Iγ computation storage circuit 43, the probabilities IR(Y_t|00), IR(Y_t|01) IR(Y_t|10), IR(Y_t|11) of received values Y_t for respective states n are outputted from the ROMs 302a to 302d from one received value Y_t to another, such that Iγ[00], Iγ[01], Iγ[10], Iγ[11] of respective branches associated with the outputs [00], [01], [10], [11] on the trellis are obtained from the adders 303a to 303d. These probabilities Iγ[00] to Iγ[11] are sequentially stored in the RAMs 304a to 304d and read out in a pre-set sequence so as to be retrieved by the selection circuit 308 as Iγ(α) Iγ(β_1), Iγ(β_2) and Iγ(γ).

[0065] Fig.9 is a timing chart showing the management of RAMs 304a to 304d. The four RAMs 304a to 304d are run in a bank configuration, having a recording capacity of 16 bits × 4 words, for storing the output data Iγ[00] to Iγ[11] of the adders 303a to 303d, in an amount corresponding to the truncated length D. The RAMs 304a to 304d thus store the probabilities Iγ[00] to Iγ[11] sequentially cyclically (Fig.9A). In Fig.9, the probabilities Iγ[00] to Iγ[11] at time points t = 1,2,3, · · · are depicted by γ1, γ2, γ3, · · ·.

[0066] From the RAMs 304a to 304d, the probabilities Iγ[00] to Iγ[11] are read out with a delay corresponding to the time equal to twice the truncated length D, or 2D. From the selection circuit 308, these probabilities Iγ[00] to Iγ[11] are retrieved as the probability Iγ(α) to be sent to the Iα computation storage circuit 44.

[0067] The operation of reading out the probabilities Iγ[00] to Iγ[11] for 2D, directly since the end of writing of the probabilities Iγ[00] to Iγ[11] in the RAMs 304a and 304b, in a reverse sequence to the writing sequence, and the operation of reading out the probabilities Iγ[00] to Iγ[11] for 2D, directly since the end of writing of the probabilities Iγ[00] to Iγ[11] in the RAMs 304c and 304d, in a reverse sequence to the writing sequence, are effected alternately. From the selection circuit 308, the probabilities Iγ[00] to Iγ[11], thus read out, are retrieved as the probability Iγ(β_1) to be supplied to the Iβ computation storage circuit 45 (Fig.9C).

[0068] On the other hand, the operation of reading out the probabilities Iγ[00] to Iγ[11] for 2D, directly since the end of writing of the probabilities Iγ[00] to Iγ[11] in the RAMs 304b and 304c, in a reverse sequence to the writing sequence, and the operation of reading out the probabilities Iγ[00] to Iγ[11] for 2D, directly since the end of writing of the probabilities Iγ[00] to Iγ[11] in the RAMs 304d and 304a, in a reverse sequence to the writing sequence, are effected alternately. From the selection circuit 308, the probabilities Iγ[00] to Iγ[11], thus read out, are retrieved as the probability I$\gamma(\beta_2)$ to be supplied to the Iβ computation storage circuit 45 (Fig.9D).

[0069] After lapse of 2D since the writing of the probabilities Iγ[00] to Iγ[11] for one truncated length D in the RAMs 304a to 304d, these probabilities Iγ[00] to Iγ[11] are read out in the reverse order to the writing order. From the selection circuit 308, these probabilities Iγ[00] to Iγ[11] are retrieved as the probability I$\gamma(\lambda)$ to be supplied to the soft output computation circuit 46 (Fig.9E).

[0070] Fig.10 shows the configuration of the Iα computation storage circuit 44. This Iα computation storage circuit 44 includes input terminals 401, 402, fed with the probability I$\gamma(\alpha)$ and with the control signal Scγ, respectively, an Iα computing circuit 403 for computing Iα in accordance with the equation (16), using the probability I$\gamma(\alpha)$ one time point ahead as set on a register 405, and a selector 404 for selecting Iα outputted by the Iα computing circuit 403 or the initial value Iα_0 to set the selected value in the register 405.

[0071] The selector 404 selects the initial value Iα_0 only at the time of initialization, based on the control signal Scγ, while selecting output data of the Iα computing circuit 403 at other time points. This initialization occurs at a time point before the time of beginning of the outputting of the probability I$\gamma(\alpha)$ by the Iγ computation storage circuit 43. As for the initial value Iα_0, log 1 (= 0) and log 0(= $-\infty$)are given as the value at the state 0 and as the value for other states, respectively, if the start point of the encoding is known on the part of a receiver. If the start point of the encoding is not known on the part of a receiver, log 1/M, herein log 1/4, is given for all states if definition is to be followed. In actuality, the same value may be used for all states, such that 0 may be given to all states, as an example.

[0072] Fig.11 shows the structure of the Iα computing circuit 403 including four addition comparison selection circuits 411a to 411d. Iγ_t[00] to Iγ_t[11] and Iα_{t-1}[0] to Iα_{t-1}[3] one time point ahead are distributed to the addition comparison selection circuits 411a to 411d based on the transitions on the trellis. Each addition comparison selection circuit 411a to 411d selects a larger one of two results of addition of Iα and Iγ to find Iα in each state at the next time point.

[0073] The addition comparison selection circuit 411a is fed with Iγ[00] and Iγ_t[11], while being fed with Iγ_{t-1}[0] and Iγ_{t-1}[2], to compute the probability Iαt(0) for the state 0. The addition comparison selection circuit 411b is fed with Iγ_t[11] and Iγ_t[00], while being fed with Iγ_{t-1}[2], to compute the probability Iαt(1) for the state 1.

[0074] The addition comparison selection circuit 411c is fed with Iγ_t[10] and Iγ_t[01], while being fed with Iα_{t-1}[1] and Iα_{t-1}[3], to compute the probability Iα_t(2) for the state 2. The addition comparison selection circuit 411d is fed with Iγ_t[01] and Iγ_t[10], while being fed with Iγ_{t-1}[1]and Iγ_{t-1}[3], to compute the probability Iα_t(3) for the state 3.

[0075] The addition comparison selection circuits 411a to 411d are configured in common as shown in Fig. 12. The addition comparison selection circuit 411a will now be explained subsequently. Iγ_t[00] (probability of a branch getting to the state 0, indicated by a broken line in Fig.3) and Iα_{t-1}[0] (probability of a branch getting to the state 0 one time point ahead in Fig.3) are summed together by an adder 421. Also, Iγt [11] (probability to the state 0 by a solid line in Fig.5) and Iα_{t-1}(2) (probability to the state 2 one time point ahead in Fig.5) are summed together by the adder 422. The results of addition by the adders 421, 422 are compared to each other in a comparator circuit 423 and the result of addition by the adder 421 or that by the adder 421, whichever is larger, is retrieved as the probability Iα_t(0). Although not explained specifically, the same operations are preformed for the addition comparison selection circuits 411b to 411d.

[0076] Reverting to Fig.10, the Iα computation storage circuit 44 includes RAMs 406, 407 for sequentially storing the probabilities Iα(0) to Iα(3), outputted by the register 405, in accordance with the control signal Scα to output the stored data in a pre-set sequence, a selection circuit 408 for selectively retrieving Iα outputted by these RAMs 406, 407 in accordance with the control signal Scα to convert it into I$\gamma(\lambda)$, and an output terminal 409 for outputting this I$\alpha(\lambda)$. If the number of bits of Iα is 8, the number of bits of the probabilities Iαt(0) to Iαt(3) is 32. These 32 bits are stored as one word in the RAMs 406, 407.

[0077] The Iα computation storage circuit 44 is initialized at a time point directly before start of outputting of the probability I$\gamma(\alpha)$ by the Iγ computation storage circuit 43 (see Fig.9B and Fig.13A). By this initialization, an initial value Iα_0 is selected by the selector 404 and the initial value thus selected (Iα_0[00] to Iα_0[11]) is set in the register 405 (Fig.13B). As from the next following clock period, the Iγ computation storage circuit 403 sequentially computes Iαt of the next time point (Fig.13B) by the Iα computing circuit 403, from the probability I$\gamma(\alpha)$ sent from the Iγ computation storage circuit 43 and the probability Iα_{t-1} outputted by the register 405 to store this Iα_t again in the register 405. In Fig.13, the probabilities Iα(0) to Iα(3) associated with the time points $t = 1, 2, 3, \cdots$ are indicated by $\alpha_1, \alpha_2, \alpha_3, \cdots$, respectively.

[0078] Figs.10C, D indicate management of RAMs 406, 407. These two RAMs 406, 407 operate in a bank configuration having a storage capacity of 32 bits x 4 words in order to store output data of the register 405, that is the probabilities Iα(0) to Iα(3), for a length corresponding to the truncated length, in order to store the probabilities Iα(0) to Iα(3) sequentially cyclically (Fig.13C).

[0079] The operation of reading the probabilities Iα(0) to Iα(3) for the truncated length D, since the time these probabilities are written in the RAM 406, in a reversed order from the writing order, and the operation of reading the probabilities Iα(0) to Iα(3) for the truncated length D, since the time these probabilities are written in the RAM 407, in a reversed order from the writing order, are performed alternately. From the selection circuit 408, the probabilities Iα(0) to Iα(3), thus read out, are retrieved as the probability Iα(λ) to be supplied to the soft output computation circuit 46 (Fig.13D).

[0080] Fig.14 shows the configuration of the Iβ computation storage circuit 45. This Iβ computation storage circuit 45 includes input terminals 501, 502, 503, fed with the probabilities Iγ(β_1), Iγ(β_2) and the control signal Scβ, respectively, an Iβ computation circuit 504 for computing Iβ in accordance with the equation (17) using Iγ(β_1) supplied to the input terminal 501 and the probability Iβ set in the register 506 and a selector 505 for selecting one of Iβ outputted by the Iβ computation circuit 504 or the initial value Iβa to set the selected value in the register 506.

[0081] The selector 505 selects the initial value Iβa only at the initializing time point, based on the control signal Scβ, while selecting output data of the Iβ computation circuit 504 at other time points. The initialization occurs at time directly before start of the outputting of the probability Iγ(β_1) by the Iγ computation storage circuit 43 and subsequently at an interval of 2D, D being a truncated length. As the initial value Iβa, usually the same value for all states, such as 0 or log 1/M, herein log 1/4, is accorded. However, when decoding the terminated code, log 1 (= 0) and log 0 (= -∞) are accorded as the value for the terminating state and other states, respectively.

[0082] Fig.15 shows the configuration of the Iβ computation circuit 504 made up of four addition comparison selection circuits 511a to 511d. $I\gamma_t$[00] to $I\gamma_t$[11] and $I\beta_t(0)$ to $I\beta_t(3)$ are distributed to the addition comparison selection circuits 511a to 511d based on transitions on the trellis. The addition comparison selection circuits 511a to 511d select a larger one of two results of addition of Iβ and Iγ to find Iβ in each state for the previous time point.

[0083] The addition comparison selection circuit 511a is fed with $I\gamma_t$[00], $I\gamma_t$[11] and with $I\beta_t(0)$, $I\beta_t(1)$ to compute the probability $I\beta_{t-1}(0)$ for the state 0. The addition comparison selection circuit 511b is fed with $I\gamma_t$[10], $I\gamma_t$[01] and with $I\beta_t(2)$, $I\beta_t(3)$ to compute the probability $I\beta_{t-1}(1)$ for the state 1.

[0084] The addition comparison selection circuit 511c is fed with $I\gamma_t$[11], $I\gamma_t$[00] and with $I\beta_t(0)$, $I\beta_t(1)$ to compute the probability $I\beta_{t-1}(2)$ for the state 2. The addition comparison selection circuit 511d is fed with $I\gamma_t$[01], $I\gamma_t$[10] and with $I\beta_t(2)$, $I\beta_t(3)$ to compute the probability $I\beta_{t-1}(3)$ for the state 3.

[0085] The addition comparison selection circuits 511a to 511d are configured in common as shown in Fig. 16. The following description is made on the addition comparison selection circuit 511a. $I\gamma_t$[00] (probability of a branch getting to the state 0, indicated by a broken line in Fig.3) and $I\beta_t$[0] (probability of getting to the state 0 by retrograding down the time axis as from the terminal point of the truncated length in Fig.5) are summed together by an adder 521. Also, $I\gamma_t$[11] (probability of a branch getting to the state 0, indicated by a solid line in Fig.5) and $I\beta_t$[1] (probability of getting to the state 1 by retrograding down the time axis as from the terminal point of the truncated length in Fig.5) are summed together by an adder 522. The results of addition by the adders 521, 522 are compared to each other in a comparator circuit 523 and the result of addition by the adder 521 or that by the adder 522, whichever is larger, is retrieved as the probability $I\beta_{t-1}(0)$. Although not explained specifically, the same operations are preformed for the addition comparison selection circuits 511b to 511d.

[0086] Reverting to Fig.14, the Iβ computation storage circuit 45 includes an Iβ computation circuit 507 for computing Iβ, in accordance with the equation (17), using the probability Iγ(β_2) sent to the input terminal 502 and the probability Iβ set in the register 509 and a selector 508 for selecting Iβ outputted by the Iβ computation circuit 507 or the initial value Iβb to set the selected value in the register 509.

[0087] The selector 508 selects the initial value Iβb only at the initializing time point based on the control signal Scβ, while selecting output data of the Iβ computation circuit 507 at other time points. The initialization occurs at time directly before start of the outputting of the probability Iγ(β_2) by the Iγ computation storage circuit 43 and subsequently at an interval of 2D, D being a truncated length. The initial value Iβb is set similarly to Iβa. Although not explained specifically, the Iβ computation circuit 507 is configured similarly to the Iβ computation circuit 504 (Figs.15 and 16).

[0088] The Iβ computation storage circuit 45 includes a selection circuit 510 for selectively retrieving the probabilities Iβ(0) to Iβ(3), outputted by the registers 506, 509, in accordance with the control signal Scβ, to convert it into Iβ(λ), and an output terminal 512 for outputting this Iβ(λ). If the number of bits of Iβ is 8, the number of bits of the probabilities $I\beta_{t-1}(0)$ to $I\beta_{t-1}(3)$ outputted by the Iβ computation circuits 504, 507 is 32.

[0089] The Iβ computation storage circuit 45, shown in Fig.14, initializes the register 506 at a time point directly before starting the outputting of the probability Iγ(β_1) by the Iγ computation storage circuit 43 (Figs.9C and 17A) and subsequently at an interval of 2D. In this initialization, the selector 505 selects the initial value Iβa, which is set in the register 506. From the next clock period on, the Iβ computation circuit 504 sequentially computes $I\beta_{t-1}$ of the previous time point, from the probability Iγ(β_1) supplied by the Iγ computation storage circuit 43 and $I\beta_t$ outputted by the register 506, this $I\beta_{t-1}$ being again stored in the register 506 and outputted at the next clock time (Fig.17C). Meanwhile, Fig.17 shows the probabilities Iβ(0) to Iβ(3), associated with the time points t = 1, 2, 3, · · ·, with β_1, β_2, β_3, · · ·, respectively.

[0090] The register 509 is initialized at a time directly previous to the start of the outputting of the probability Iγ(β2) by the Iγ computation storage circuit 143(Figs.9D and 17B) and subsequently at an interval of 2D. In this initialization, the selector 508 selects the initial value Iβb, which is set in the register 506. From the next clock period on, the Iβ computation circuit 507 sequentially computes $I\beta_{t-1}$ of the previous time point, from the probability Iγ(β_2) supplied by the Iγ computation storage circuit 43 and $I\beta_t$ outputted by the register 509, this $I\beta_{t-1}$ being again stored in the register 509 and outputted at the next clock time (Fig.17D). The selection circuit 510 selectively retrieves outputs of the registers 506, 509 to derive the probability $I\beta(\gamma)$ to be supplied to the soft output computation circuit 46, as shown in Fig.17E.

[0091] Fig.18 shows the configuration of the soft output computation circuit 46, which is made up of input terminals 601, 602, 603 fed with the probabilities $I\alpha(\lambda)$, $I\beta(\lambda)$ and $I\gamma(\lambda)$, respectively, an $I\lambda_1$ computing circuit 604 and an $I\lambda_0$ computing circuit 605 for computing the first and second terms of the right side of the equation (19), a subtractor 606 for subtracting the output $I\lambda_0$ of the computing circuit 605 from the output $I\lambda_1$ of the computing circuit 604 to obtain $I\lambda_1$ of the equation (19), a last-in first-out (LIFO) memory 607 for chronologically re-arraying $I\lambda_1$ outputted by the subtractor 606 to output the re-arrayed data and an output terminal 608 for outputting the soft output $I\gamma_t$.

[0092] Fig.19 shows the configuration of the $I\lambda_1$ computing circuit 604. This $I\lambda_1$ computing circuit 604 includes four adders 604a to 604d and a maximum value selection circuit 604e. On the adders 604a to 604d, signals are distributed in the following fashion based on the status transitions on the trellis. That is, $I\alpha_{t-1}(0)$, $I\beta_t(1)$, $I\gamma_t[11]$ are supplied to the adder 604a, $I\alpha_{t-1}(1)$, $I\beta_t(2)$, $I\gamma_t[10]$ are supplied to the adder 604b, $I\alpha_{t-1}(2)$, $I\beta_t(0)$, $I\gamma_t[11]$ are supplied to the adder 604c and $I\alpha_{t-1}(3)$, $I\beta_t(3)$, $I\gamma_t[10]$ are supplied to the adder 604d.

[0093] The maximum value of the results of addition by the adders 604a to 604d is selected by the maximum value selection circuit 604e and outputted as $I\lambda_1$.

[0094] Fig.20 similarly shows the configuration of the $I\lambda_0$ computing circuit 605.

[0095] This $I\lambda_2$ computing circuit 605 includes four adders 605a to 605d and a maximum value selection circuit 605e. On the adders 605a to 605d, signals are distributed in the following fashion based on the status transitions on the trellis. That is, $I\alpha_{t-1}(0)$, $I\beta_t(0)$, $I\gamma_t[00]$ are supplied to the adder 605a, $I\alpha_{t-1}(1)$, $I\beta_t(3)$, $I\gamma_t[01]$ are supplied to the adder 604b, $I\alpha_{t-1}(2)$, $I\beta_t(1)$, $I\gamma_t[00]$ are supplied to the adder 605c and $I\alpha_{t-1}(3)$, $I\beta_t(2)$, $I\gamma_t[01]$ are supplied to the adder 605d. The maximum value of the results of addition by the adders 605a to 605d is selected by the maximum value selection circuit 605e and outputted as $I\lambda_0$.

[0096] To the input terminals 601, 602, 603 of the soft output computation circuit 46 are fed the probabilities $I\alpha(\lambda)$, $I\beta(\lambda)$ and $I\gamma(\lambda)$, respectively (Figs.21A, B and C). At each clock period, the $I\lambda_1$ computing circuit 604 computes the first term of the right side of the equation (19) to obtain $I\lambda_1$, while the $I\lambda_0$ computing circuit 605 computes the second term of the right side of the equation (19) to obtain $I\lambda_1$, so that the subtractor 606 outputs $I\lambda_t$ at each time point t (Fig.21D). The subtractor 606 sequentially outputs $I_{\lambda 1}$ which is fed to the LIFO memory 607 and chronologically re-arrayed so as to be outputted as re-arrayed soft output $I\lambda_t$. In Fig.21, the soft output Iλt associated with the time points t = 1, 2, 3, • • •, is denoted as inputs λ_1, λ_2, λ_3, • • •,

[0097] Referring to the drawings, the memory management by the controller 41 in the decoder 4 is explained in further detail. Figs.22A to 22D show the contents of the memory management by chronologically showing the stored contents and outputs of the RAMs 304a to 304d, register 405, RAMs 406, 407 and registers 506, 509. In the RAMs 304a to 304d and RAMs 406, 407, arrows ↓ and ↑ indicate writing in a specified address and readout from a specified address, respectively.

[0098] In Figs.22A to 22D, the following operations (1) to (6) are simultaneously executed at, for example, t = 13:

(1) $I\gamma_{13}$ is stored in the RAM 304d;

(2) $I\alpha_6$ then is found on the basis of $I\alpha_4$ outputted by the register 405 and $I\gamma_5$ outputted by the RAM 304b to store $I\alpha_5$ thus found again in the register 405;

(3) $I\alpha_4$ found at the previous time point to the outputting from the register 506 is stored in the RAM 407;

(4) $I\beta_3$ is found on the basis of $I\beta_4$ outputted by the register 506 and $I\gamma_4$ outputted by the RAM 304a to store $I\beta_4$ thus found again in the register 506;

(5) $I\beta_{11}$ is found on the basis of $I\beta_{12}$ outputted by the register 509 and $I\gamma_{12}$ outputted by the RAM 304c to find $I\beta_3$ which again is stored in the register 506; and

(6) $I\alpha_4$ is found on the basis of $I\beta_4$ outputted by the register 506, $I\gamma_4$ outputted by the RAM 304a and $I\alpha_3$ outputted by the RAM 406.

[0099] Similar operations are performed for other time points. By iteration of these operations, λ_t can be found one at a time. However, since λ_t is found in this method in the reverse order with respect to the intrinsic chronological order, the soft output λ_t is first re-arrayed in the intrinsic chronological order, by exploiting the LIFO memory 607, as described above. The re-arrayed soft output is then outputted. Fig.23 shows the timing chart for t = 13 until t = 20 if the memory management which is based on the above-described operations.

[0100] In the preferred embodiment, computation of $I\beta$ in the truncated length (computation by the $I\beta$ computation circuit 507 of Fig.14) and the computation of $I\beta$ retrograding down the time axis by not less than the truncated length (computation by the $I\beta$ computation circuit 504 of Fig.14) are carried out in parallel, so that the computation of $I\beta$ per clock is the number of states x 2. Thus, the volume of computation can be reduced significantly in comparison with the conventional SW-Max-Log-BCJR algorithm. On the other hand, the accessing to each memory per clock only once suffices. Therefore, in the preferred embodiment, the decoding operation of the convolutional codes can be executed speedily.

[0101] Meanwhile, since the memory management in the preferred embodiment is not performed in dependence upon the method for computation of $I\alpha$, $I\beta$ or $I\gamma$, it is possible to use a method other than the method of referring to the ROM table as the $I\gamma$ computing method.

[0102] The SW-Max-Log-BCJR algorithm may be mounted by assembling the correction shown by the equation (20) in the computing circuits shown in Figs. 11, 15, 19 and 20. In the following description, a soft output $I\lambda t$ is to be found in accordance with the SW-Log-BCJR algorithm is to be found.

[0103] As an example, it is assumed that the correction shown in Fig.20 is to be assembled into the circuit of Fig.11.

[0104] For assembling the correction shown by the equation (20), it is necessary to replace the structure shown in Fig.12 by the structure shown in Fig.24 in each of the addition comparison selection circuits 411a to 411d. In Fig.24, the parts corresponding to those of Fig.9 are depicted by the same reference numerals.

[0105] The addition comparison selection circuit 411a is now explained. $I\gamma_t[00]$ and $I\gamma_{t-1}(0)$ are summed in the adder 421, while $I\gamma_t[11]$ and $I\gamma_{t-1}(2)$ are summed in the adder 422. A subtractor 426 effects subtraction of the result of addition of x of the adder 421 and the result of addition y of the adder 422 and the result of subtraction (x - y) is sent to a positive/negative decision circuit 427, from which signals "1" and "0" are outputted if the result of subtraction (x - y) is not less than 0 and less than 0, respectively.

[0106] An output signal of the positive/negative decision circuit 427 is sent to the selector 424 as a selection signal SEL. From the selector 424, the result of addition of x of the adder 421 or the result of addition y of the adder 422 is retrieved depending on whether the selection signal SEL is "1" or "0", respectively. The result is that the result of addition x or y of the adder 421 or 422, whichever is larger, is selectively taken out to execute the computation corresponding to the first term of the right side of the equation (20).

[0107] The result of subtraction (x - y) of the subtractor 426 is sent to an absolute value computing circuit 428 to compute an absolute value |x - y|. This absolute value |x - y| is sent as a readout address signal to a ROM 429 constituting the table. From this ROM 429, $\log(1 + e^{-|x-y|})$, as the second term of the right side is obtained in the equation (20). An adder 430 sums an output signal max(x - y) of the selector 424 to an output signal $\log(1+ e^{-|x-y|})$ of the ROM 429 to output the result of addition as a probability $I\alpha_t(0)$ which is based on the SW-Log- BCJR algorithm. The above holds for the addition comparison selection circuits 411b to 411d, although these circuits are not explained specifically.

[0108] The foregoing description has been made of assembling of the correction shown in the equation (20) into the circuit of Fig 11. The correction indicated in the equation (20) can be made similarly for the circuits shown in Figs.15, 19 and 20 to enable the mounting of the SW-Log- BCJR algorithm.

[0109] In the above-described embodiment, the constraint length = 3 and the truncated length = 4. However, the constraint length and the truncated length may be of any suitable value without being limited to the above values. On the other hand, the RAM configuration may be variegated such as by replacing the RAMs 304a to 304d by two dual port RAMs or by replacing the RAMs 406, 407 by a sole dual port RAM, or by employing a multi-port RAM in place of a single-port RAM, even though the memory read/write contents remain unchanged.

[0110] The memory management may be variegated, such as by storing $I\beta$ instead of $I\alpha$ in a RAM. Although the SW-Max- BCJR algorithm and the SW-Log- BCJR algorithm are used in the above embodiment, any other suited soft output decoding algorithm may be used by way of further modifications.

[0111] In the above-described respectively embodiments, the probabilities $I\gamma[00]$ to $I\gamma[11]$ are read out with a delay corresponding to the truncated length D times 2, or 2D, from the RAMs 304a to 304d to decode the soft output. However, the delaying time for the probability information not less than the truncated length D suffices, without being limited to the truncated length D times 2, or 2D.

[0112] For example, it is also possible to extend a RAM 308 to the $I\gamma$ computation storage circuit 43 shown in Fig.8 to delay the probability information for a time duration corresponding to the truncated length D times 3, or 3D.

[0113] The $I\gamma$ computation storage circuit 43, configured as shown in Fig.25, sequentially stores the probabilities $I\gamma[00]$, $I\gamma[01]$, $I\gamma[10]$, $I\gamma[11]$ of the respective branches, associated with outputs [00], [01], [10], [11] on the trellis obtained by the adders 303a to 303, in the RAMs 304a to 304e (Fig.26A), to output probabilities $I\gamma[00]$ to $I\gamma[11]$ held with delay for a period corresponding to the truncated length D times 3, or 3D. The selection circuit 308 thus outputs the data $I\gamma[00]$ to $I\gamma[11]$ after delay as the probabilities $I\gamma(\alpha)$ for the $I\gamma$ computation storage circuit 43 (Figs.26B, C).

[0114] The RAMs 304a to 304e and the selection circuit 308 partitions the probabilities $I\gamma[00]$ to $I\gamma[11]$ every truncated length D and sets a reference time point at a time point which has elapsed the truncated length D along the time axis if the direction as from the terminal point of each truncated length D. If the probabilities $I\gamma[00]$ to $I\gamma[11]$ are stored in a volume corresponding to truncated length D times 2 up to these reference time points, the RAMs 304a to 304e and the selection circuit 308 output these probabilities Iγ[00] to Iγ[11] in a reverse order from the input sequence. Thus, the RAMs 304a to 304e and the selection circuit 308 output the probabilities Iγ[00] to Iγ[11] as the first probability Iγ(β_1) for the Iβ computation storage circuit 45 (Figs.26D, E), while outputting the probabilities Iγ[00] to Iγ[11], constituted by the probability Iγ delayed by the truncated length D from the first probability Iγ(β_1) in a similar sequence as the second probability Iγ(β2) for the Iβ computation storage circuit 45 (Figs.26F and G).

[0115] For the probability Iγ(λ) for the soft output computation circuit 46, the RAMs 304a to 304e and the selection circuit 308 outputs the probabilities Iγ[00] to Iγ[11], delayed a pre-set time in a sequence for the Iα computation storage circuit 44. Thus, the Iγ computation storage circuit 43 outputs the first probability Iγ in a sequence associated with the processing in the Iα computation storage circuit 44, Iβ computation storage circuit 45 and in the soft output computation circuit 46.

[0116] Referring to Fig.27, the Iβ computation storage circuit 45 sends the probabilities $\beta_{t-1}(0)$ to $\beta_{t-1}(3)$, computed by the Iβ computation circuits 504, 507, to the selection circuit 510 via RAMs 513, 514.

[0117] In the Iβ computation storage circuit 45, shown in Fig.27, the Iβ computation circuits 504, 507 compute, from the first probability Iγ(β_1) and the second probability Iγ(β_2) outputted by the Iγ computation storage circuit 43, the probabilities $\beta_{t-1}(0)$ to $\beta_{t-1}(3)$, retrograding to each state of the received value β_t, based on the probabilities $\beta_{t-1}(0)$ to $\beta_{t-1}(3)$, inputted from the Iβ computation circuits 504, 507 and from the registers 506, with a lead of one clock period.

[0118] That is, in this Iβ computation storage circuit 45, the selectors 505, 508 selectively output, under control by the control signal Scβ, the probabilities $\beta_t(0)$ to $\beta_t(3)$ or the initial values Iβa, Iβb to the registers 506, 507, these probabilities βt(0) to Yt(3) being outputted by the Iβ computation circuits 504, 507 with a lead of one clock period.

[0119] As the initial values Iβa, Iβb, the same value, such as 0 or log 1/M, herein log 1/4, is usually given as described above. When decoding the terminated code, log 1 (= 0) and log 0 (= -∞) are given as the value in the terminated state and as the value in the other state, respectively.

[0120] Referring to Fig.28, the selectors 505, 508 selectively output the initial values Iβa, Iβb to the registers 508, 509 at a timing temporally ahead by one clock period at which the repetition of the first probability Iγ(β_1) and the second probability Iγ(β_2) is changed over with the time of the truncated length D times 2 or 2D as a unit in association with the repetition of the first probability Iγ(β_1) and the second probability Iγ(β_2) retrograding down time axis with the time of 2D as a unit (Figs.28A, 28B, 28D and 28E). For other timings, probabilities $\beta_t(0)$ to $\beta_t(3)$, outputted by the Iβ computation circuits 504, 507, with a lead of one clock period, are outputted selectively.

[0121] The RAMs 513, 514 are formed by a bank configuration having a capacity of storing the probabilities $\beta_{t-1}(0)$ to $\beta_{t-1}(3)$ in an amount corresponding to the truncated length. From the probabilities $\beta_{t-1}(0)$ to $\beta_{t-1}(3)$, outputted by the Iβ computation circuits 504, 507 with a shift equal to the truncated length times 2, that is 2D, obtained in terms of a truncated length times 2, that is 2D, as a period (Figs.28B to 28D), the probabilities $\beta_{t-1}(0)$ to $\beta_{t-1}(3)$, obtained for the latter truncated length, are inputted sequentially cyclically. Thus, from the two channels of the probabilities βt-1(0) to βt-1(3), computed from the initial values Iβa, Iβb, as reference, the RAMs 513, 514 selectively retrieve sufficiently reliable portions to output the so-stored $\beta_{t-1}(0)$ to $\beta_{t-1}(3)$ chronologically.

[0122] The Iβ computation storage circuit 45 processes the first probability Iγ simultaneously in parallel by plural channels, within the extent of the truncated length in question as from each reference time point as set for at least the first probability Iγ to compute the third probability Iβ by plural channels to output selectively the probability Iβ of the truncated length by these plural channels of the probabilities Iγ. Meanwhile, in the preferred embodiment, these reference time points are set at the terminal point of the truncated length next to each truncated length.

[0123] The selection circuit 510 output the probabilities $\beta_{t-1}(0)$ to $\beta_{t-1}(3)$, alternatively outputted from the RAMs 513, 514 in this sequence in terms of the truncated length D as a unit, to the soft output computation circuit 15.

[0124] In the above structure, an input i_t is convolution-encoded with a constraint length 3, by the convolutional encoder 2, where it is converted into an output string X_t, having a number of states m equal to 4, this output string X_t being inputted to the decoder 4 over the non-storage communication route 3. In this decoder 4, the received value of the input signal is detected by analog/digital conversion and inputted to the Iγ computation storage circuit 43.

[0125] In the Iγ computation storage circuit 43, the probability of the received value β_t associated with each state is computed by the table of the ROMs 302a to 302d, associated with each state (first term of the right side of the equation (16) and summed to the initial value log Pr{i_t = 0} by the adders 303a to 303d to compute sequentially the first probability Iγ associated with each state m (Fig.26A).

[0126] The first probability Iγ, thus computed, is sequentially stored in the RAMs 304a to 304e of the bank configuration, constituted with the truncated length D as a unit, so as to be sequentially stored in the RAMs 304a to 304e of the bank structure constituted in terms of the truncated length D as a unit, and so as to be outputted chronologically to the Iγ computation storage circuit 42 after a delay corresponding to a time equal to the truncated length D times three (Fig.26C). This outputs the first probability Iγ(α) to the Iγ computation storage circuit 43 in a sequence and timing proper for the processing by the Iγ computation storage circuit 43.

[0127] If the first probability Iγ is partitioned in terms of the truncated length D as a unit, each reference time point is set at a time point corresponding to the truncated length as from each truncated length D, and the first probability Iγ is stored in the RAMs 304a to 304e as from each truncated length to the corresponding reference time point, the partitioned portions of the first probability Iγ is outputted to the Iβ computation storage circuit 45 in a sequence retrograding down the time axis as from each reference time point (Figs.26D to G). Since the reference time point is set at en end time point of the next following truncated length, the partitioned portion of the first probability Iγ as from the next following truncated length up to the corresponding reference time point start to be outputted, before completion of the outputting of the partitioned portion of the first probability Iγ as from a given truncated length up to the reference time point. Thus, the first probability Iγ is outputted to the Iβ computation storage circuit 45 in a sequence retrograding down the time axis by two routes shifted in contents and timing by one truncated length (see Figs.26D to G). This outputs two routes of the second probability I$\gamma(\beta_1)$ in a sequence and at a timing suited to the processing in the Iβ computation storage circuit 45.

[0128] On the other hand, the first probability Iγ is chronologically outputted to the soft output computation circuit 46, 44,with a pre-set time delay, in the same way as it is outputted tot the Iα computation storage circuit 44.

[0129] The Iα computation storage circuit 44 computes the second probability $\alpha(\lambda)$ getting to each state from the first probability Iγ in a direction along the time axis every received value to output the computed second probability $\alpha(\lambda)$ in a sequence and timing suited to processing in the soft output computation circuit 46.

[0130] In the Iβ computation storage circuit 45, the two routes of the input probabilities I$\gamma(\beta_1)$ and I$\gamma(\beta_2)$ in the retrogressive direction down the time axis are summed in the Iβ computation circuits 504, 507 to the probabilities βt of the corresponding states one clock back and a smaller one of the summed values is selected to compute the probability in the retrogressive direction down the time axis every state. By selectively feeding back the sequentially computed probabilities or the initial values Iβa, Iβb to the Iβ computation circuits 504, 507 by the RAMs 304a, 304b, the probability in the direction retrograding from the reference time point is calculated simultaneously in parallel.

[0131] The two channels of the probabilities of the truncated length are selectively stored and outputted with sufficient reliability in an area remote from the reference point. At the time of selective outputting of the probability via the RAMs 513, 514, these probabilities are re-arrayed in the sequence along the time axis reversed from that at the time of storage and are outputted to the soft output computation circuit 46. The third probability I$\beta(\lambda)$ is outputted in a sequence corresponding to the processing in the soft output computation circuit 46 along the time axis.

[0132] The soft output computation circuit 46 can execute the processing without re-arraying the input probabilities I$\alpha(\lambda)$ I$\beta(\lambda)$ and I$\gamma(\lambda)$ and without re-arraying the computed soft outputs Iγ, thus enabling the soft outputs to be obtained by a simplified structure.

[0133] Since the third probability can be computed by computation of two states x number of states, instead of by computation of the constraint length x number of states, in decoding a symbol, the processing volume can be correspondingly reduced to simplify the entire structure.

[0134] That is, the probabilities I$\alpha(\lambda)$ I$\beta(\lambda)$ and I$\gamma(\lambda)$, inputted in this manner to the soft output computation circuit 46, are summed every corresponding state. The maximum value of these probabilities is detected every time the "1" or the value "0" is entered and is processed by a subtractor 606 to give a soft output Iγ_t which is outputted in a time sequence associated with the received value β_t.

[0135] Thus, by partitioning the first probabilities Iγ every truncated length D to set the corresponding reference time point, by outputting the probability Iγ of the truncated length from each reference time point as a unit by plural channels in a direction retrograding down time axis, and by outputting the first, second and third probabilities Iγ, Iα and Iγ in a sequence proper to the next following processing, the soft output of each symbol can be obtained by the computation of two channels x number of states in place of the computation of the constraint length x number of states. At this time, the soft output computation circuit is able to compute the soft output Iγ without re-arraying Iα, Iβ or the computed soft output Iγ. This enables the soft output to be decoded by a simplified structure.

[0136] In the above-described embodiment the reference time point is set in terms of the truncated length as a unit. The present invention, however, is not limited to this embodiment since the reference time point can be optionally set to various positions. If, by the reference time point, thus set, the length as from a given reference time point up to a corresponding truncated length is longer than in the above-described embodiment, it becomes necessary to output the first probability Iγ via a corresponding number of channels and to process the outputted probabilities Iγ by corresponding channels in the Iγ computation storage circuit.

[0137] In the above-described embodiment, the soft output is computed by a truncated length equal to 4. The present invention, however, is not limited to this embodiment since the truncated length can be optionally set to any suitable value.

[0138] In the above-described embodiment, the convolutional encoding is effected with the constraint length equal to 3. The present invention, however, is not limited to this embodiment since the present invention can be applied to processing of convolutional codes by any suitable constraint length.

[0139] Also, in the above-described embodiment, the soft output is computed by the SW-Max-Log-BCJR algorithm. The present invention, however, is not limited to this embodiment since the present invention can be applied extensively to computation of the soft output by a variety of suitable soft output decoding algorithms, such as SW-Log-BCJR algorithm.

[0140] Thus, according to the present invention, in which the probability information is stored in an amount not less than the truncated length D and the updating of the probability information within the truncated length and the computation of the soft output outside the truncated length are carried out in parallel, the volume of computations per clock and the volume of accessing per memory can be reduced significantly to expedite the decoding of the convolutional codes. Moreover, by computing and processing the probabilities leading to respective states in a direction retrograding down the time axis with a length not less than the truncated length as a unit over plural channels to compute the soft output, and by outputting the computed probabilities in a sequence suited to the next following computation, a soft output decoding method and apparatus can be realized in which the soft output can be decoded by a simplified configuration.

Claims

1. A soft output decoding apparatus comprising:

   probability computing means for finding the probability information in each transition state of the convolutional codes;
   probability storage means for storing the probability information as found by said probability computing means in a recording medium; and
   soft output computing means for finding a soft output using the probability information stored in said recording medium; wherein the improvement resides in that
   said probability storage means stores the probability information in an amount not less than a truncated length on said recording medium; and in that
   the updating of the probability information within the truncated length by said probability storage means and the computation of the soft output outside the truncated length by said soft output computing means are cried out in parallel.

2. The soft output decoding apparatus according to claim 1 wherein said probability computing means and the soft output computing means compute the product processing of said probabilities by logarithmic addition processing to compute the addition processing of said probabilities by logarithmic maximum value processing.

3. The soft output decoding apparatus according to claim 1 wherein said probability computing means and the soft output computing means compute the product processing of said probabilities by logarithmic addition processing and degree-one function processing to compute the addition processing of said probabilities by logarithmic maximum value processing.

4. The soft output decoding apparatus for convolutional codes according to claim 1 wherein said probability computing means comprises:

   first computing means for computing the first probability as determined by a code output pattern and said received value;

   second computing means for chronologically computing the second probability from the start of encoding up to each state every received value based on said first probability; and

   third computing means for computing a third probability every received value from the truncation state up to each state in a sequence reversed from the chronological sequence based on said first probability;

   the probability information updated within the truncated length of said probability storage means is said third probability information.

5. The soft output decoding apparatus for convolutional codes according to claim 4 wherein said first probability computing means transiently stores said first probabilities in said probability storage means and sequentially reads out and outputs the first probabilities in a sequence corresponding to each processing in said soft output computation means; and wherein

   said third probability computing means partitions said first probabilities in terms of said pre-set truncated length as a unit, sets said reference time point along the time axis direction as from said truncated length, processes said first probabilities simultaneously in parallel by plural channels, at least in terms of a length corresponding to the corresponding truncated length as from the truncated length as from each reference time point as a unit, to compute said third probability by plural channels, selects the third probability corresponding to said truncated length from the computed third probabilities of the plural channels, to store the third probabilities corresponding to the received value transiently in said probability storage means and sequentially reads out and outputs the third probabilities in a sequence corresponding to the processing by said soft output computation means.

6. The soft output decoding apparatus for convolutional codes according to claim 5 wherein the first probabilities transiently held by said probability storage means are delayed a pre-set time in a sequence along the time axis, at least in terms of a length inclusive of said first probabilities of said truncated length corresponding to each reference time point, and wherein the first probabilities are outputted simultaneously in parallel to said third probability computing means over plural channels in a sequence retrograding down the time axis; and wherein

said first probabilities transiently stored in said probability storage means are delayed a pre-set time in a direction along time axis and outputted to said soft output computation means.

7. The soft output decoding apparatus for convolutional codes according to claim 5 wherein said reference time point is set at an end time point of the next following truncated length.

8. A soft output decoding method for convolutional codes comprising:

a first step of finding the probability information at each transition state of the convolutional codes;

a second step of storing the probability information as found in said first step in said recording medium in said first step a length not less than a truncated length; and

a third step of finding a soft output using said probability information stored in the recording medium in said second step; wherein the improvement comprises:

carrying out the updating of the probability information within the truncated length in said second step and the computation of the soft output outside the truncated length in said third step in parallel.

9. The soft output decoding method for convolutional codes according to claim 8 wherein said first step includes

a first probability computing step of sequentially computing, for each received value, the first probability as determined by an output pattern of a code and said received value;

a second probability computing step of computing, for each received value, a second probability reaching each state in a direction along the time axis, based on said first probability;

a third probability computing step of computing, for each received value, a third probability reaching each state in a direction retrograding down the time axis, as from a pre-set reference time point, based on said first probability; and

a soft output computing step of computing a soft output based on said first, second and third probabilities; wherein

in said third probability computing means, said first probability is partitioned in terms of said pre-set truncated length as a unit, said reference time point is set along the time axis direction as from said truncated length, said first probability is processed simultaneously in parallel by plural channels, at least in terms of a length corresponding to a truncated length as from each reference time point as a unit, to compute said third probabilities by plural channels, the third probability corresponding to said truncated length are selected from the computed plural channels of the third probability, and the third probability corresponding to the received value is outputted; and wherein

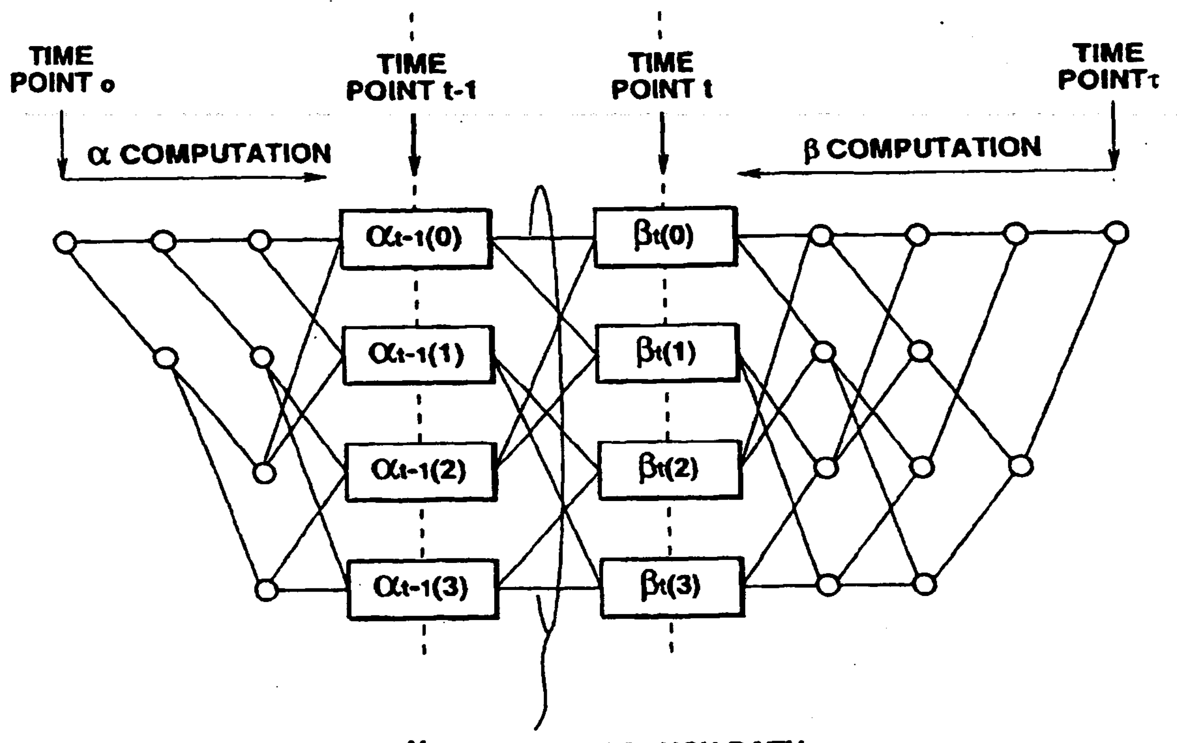
in said first probability computing step, said first probability is transiently held in probability storage means and sequentially read out and outputted in a sequence corresponding to each processing in said second and third probability computing steps and in said soft output computing step; said third probability being transiently held in probability storage means during said third probability computing step and sequentially read out and outputted in a sequence corresponding to the processing in said soft output computing step.

10. The soft output decoding method for convolutional codes according to claim 9 wherein the first probability transiently held by said probability storage means is delayed a pre-set time in a sequence along the time axis, at least in terms of a length indusive of said first probability of said truncated length corresponding to each reference time point as a unit, and wherein the first probability is outputted simultaneously in parallel to said third probability computing means over plural channels in a sequence retrograding down the time axis.

11. The soft output decoding method for convolutional codes according to claim 9 wherein said reference time point is set to an end point of the next following truncated length.

12. The soft output decoding method according to claim 9 wherein the product processing of said probabilities is computed by logarithmic addition processing in said first, second and third probability computing steps and the soft output computing step and wherein the addition processing of said probabilities is computed by logarithmic maximum value processing.

13. The soft output decoding method according to claim 9 wherein the product processing of said probabilities is computed by logarithmic addition processing in said first, second and third probability computing sub-steps and the soft output computing step and wherein the addition processing of said probabilities is computed by logarithmic maximum value processing and degree-one function processing.



COMPUTE γt FOR EACH BRANCH PATH

FIG.1

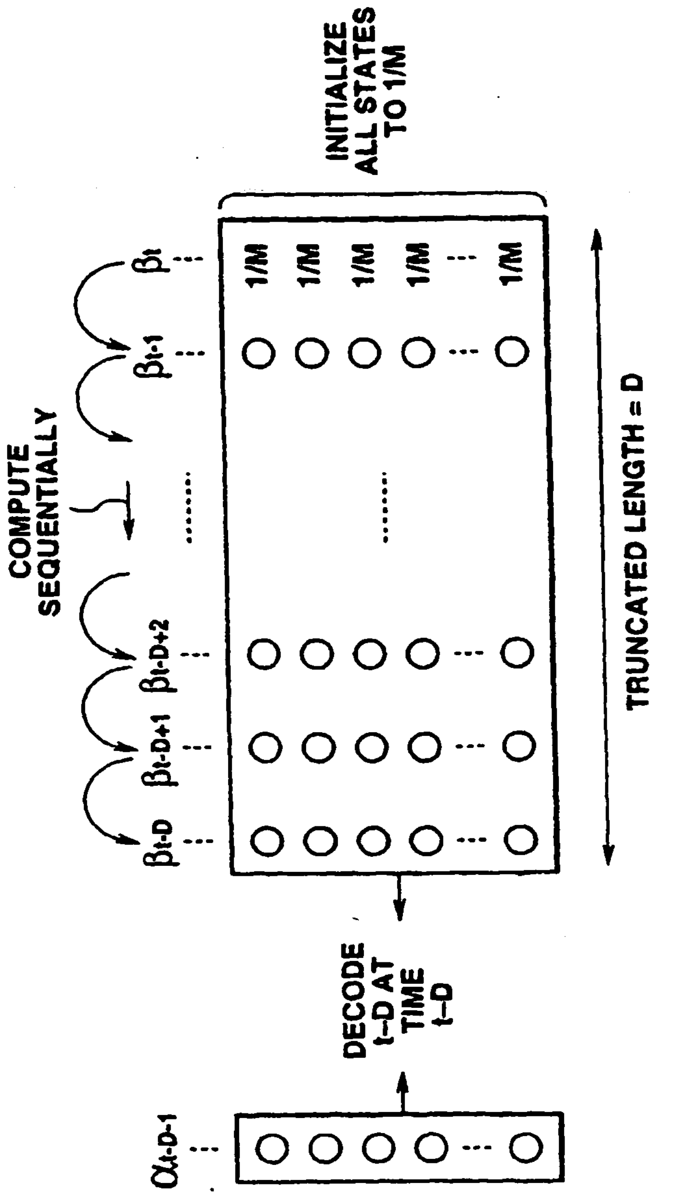

COMPUTE SEQUENTIALLY
INITIALIZE ALL STATES TO 1/M
β_{t-D}
β_{t-D+1}
β_{t-D+2}
β_{t-1}
β_t
1/M
1/M
1/M
1/M
1/M
TRUNCATED LENGTH = D
DECODE t-D AT TIME t-D
α_{t-D-1}

FIG.2

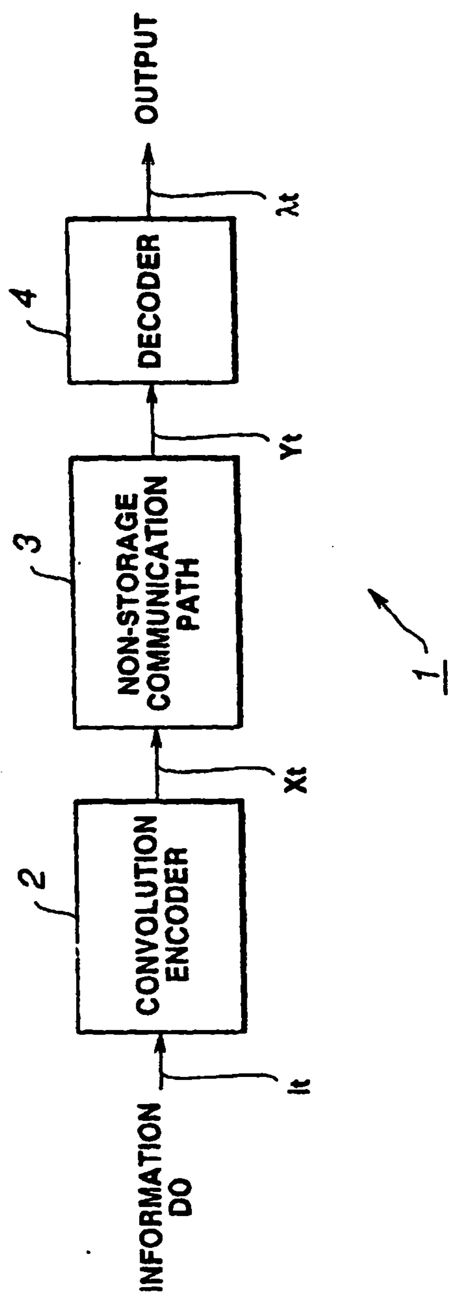

INFORMATION DO
It
2
CONVOLUTION ENCODER
Xt
3
NON-STORAGE COMMUNICATION PATH
Yt
4
DECODER
λt
OUTPUT
1

FIG.3

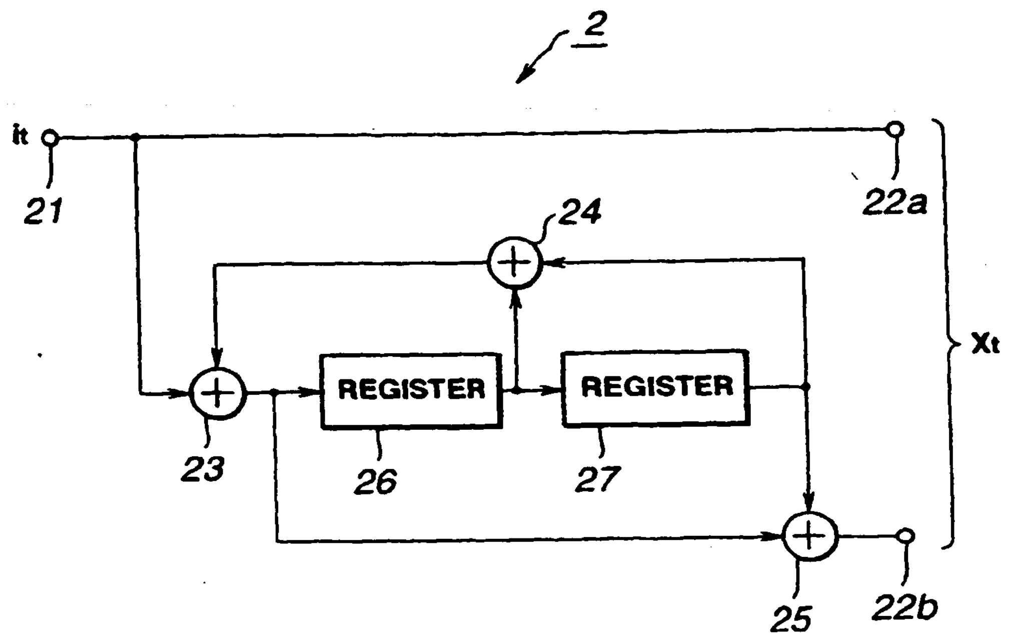


FIG.4

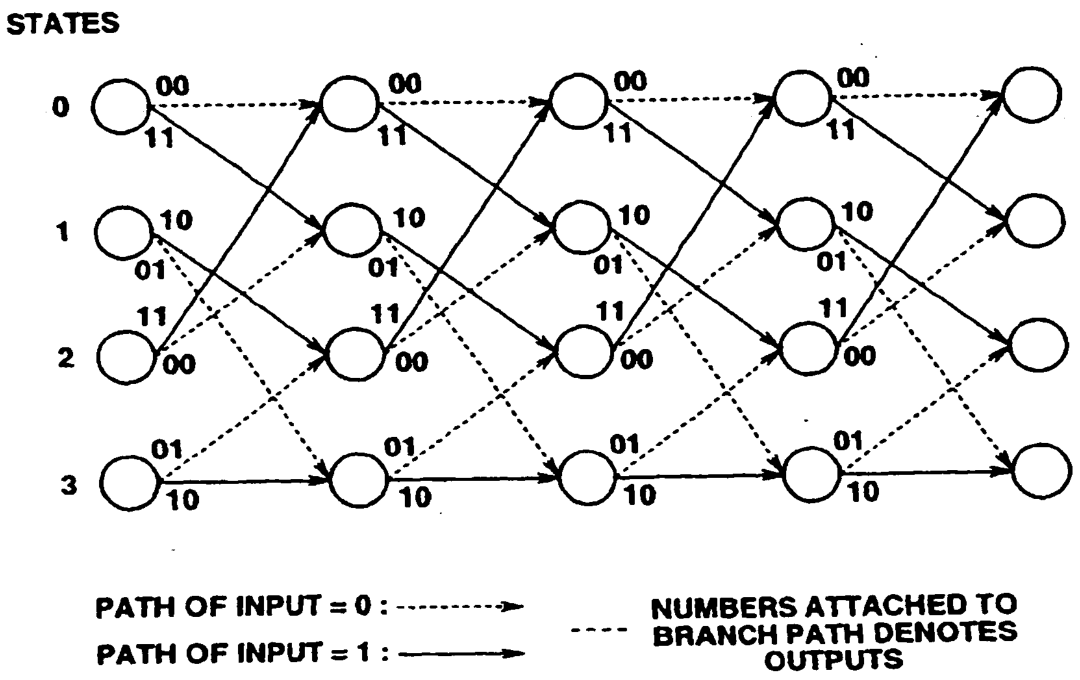

STATES
0
1
2
3
00
11
10
01
11
00
01
10
PATH OF INPUT = 0 :
PATH OF INPUT = 1 :
NUMBERS ATTACHED TO BRANCH PATH DENOTES OUTPUTS

FIG.5

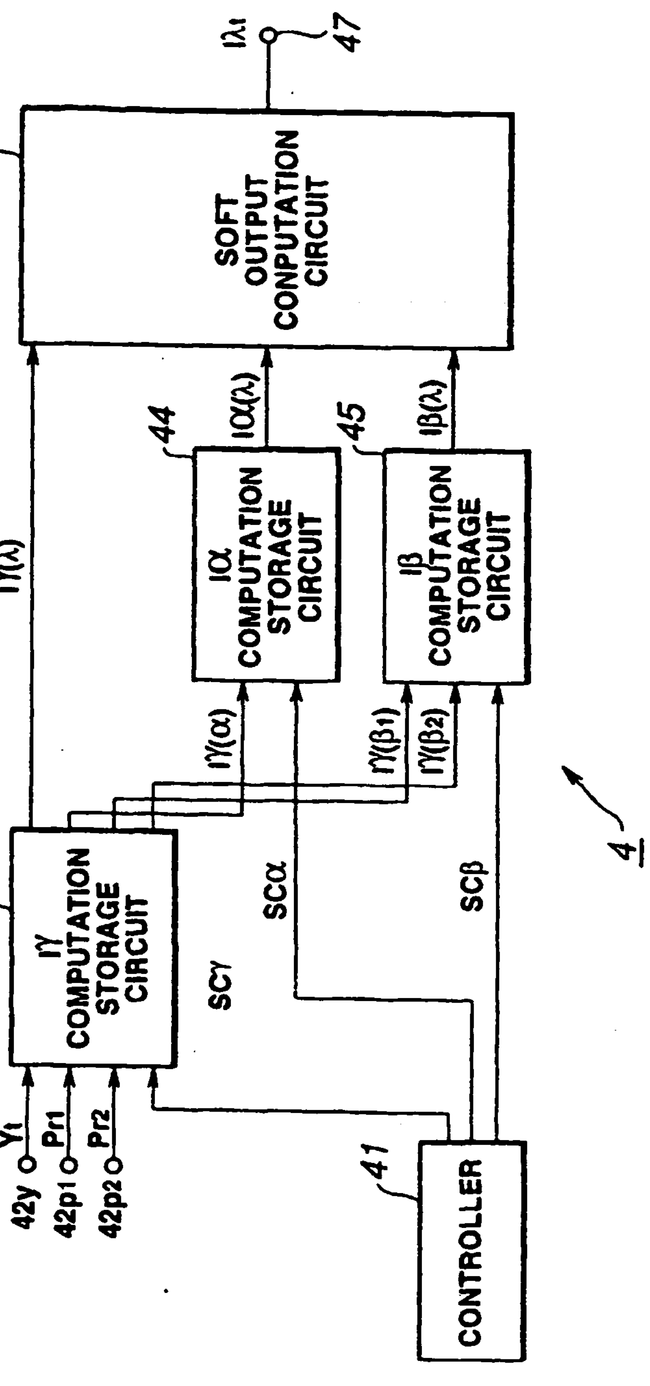

42y
Yt
42p1
Pr1
42p2
Pr2
41
CONTROLLER
43
Iγ COMPUTATION STORAGE CIRCUIT
SCγ
SCα
SCβ
Iγ(λ)
Iγ(α)
Iγ(β1)
Iγ(β2)
44
Iα COMPUTATION STORAGE CIRCUIT
45
Iβ COMPUTATION STORAGE CIRCUIT
Iα(λ)
Iβ(λ)
46
SOFT OUTPUT CONPUTATION CIRCUIT
λt
47
4

FIG.6

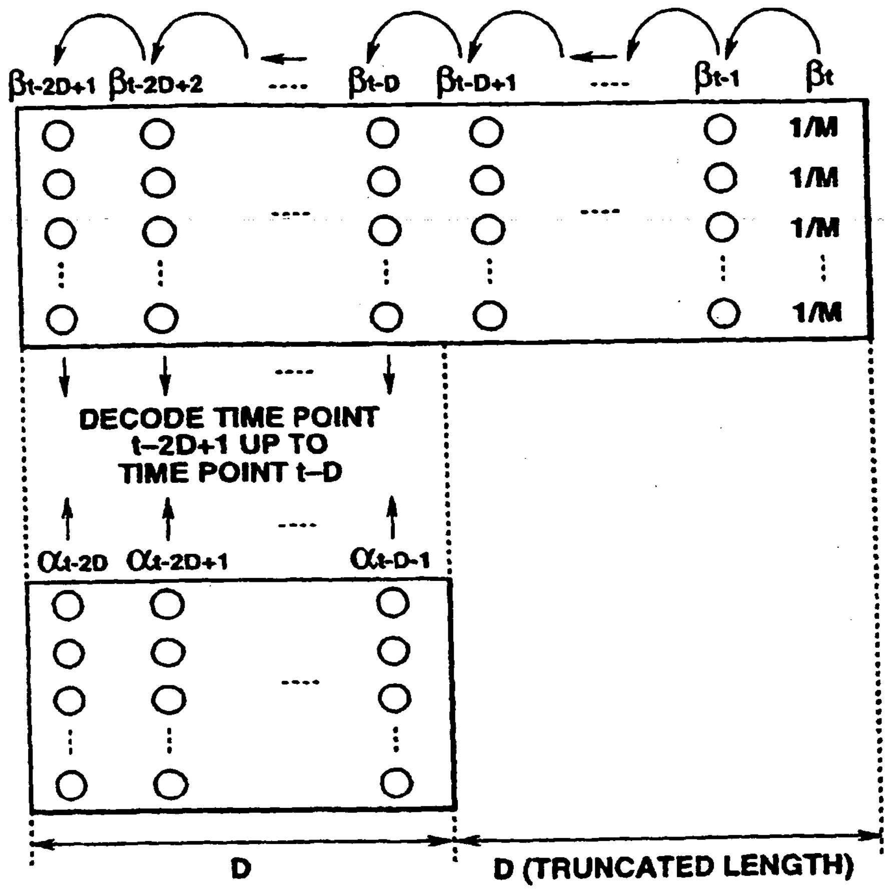

β_{t-2D+1}
β_{t-2D+2}
β_{t-D}
β_{t-D+1}
β_{t-1}
β_{t}
1/M
1/M
1/M
1/M
DECODE TIME POINT t–2D+1 UP TO TIME POINT t–D
α_{t-2D}
α_{t-2D+1}
α_{t-D-1}
D
D (TRUNCATED LENGTH)

FIG.7

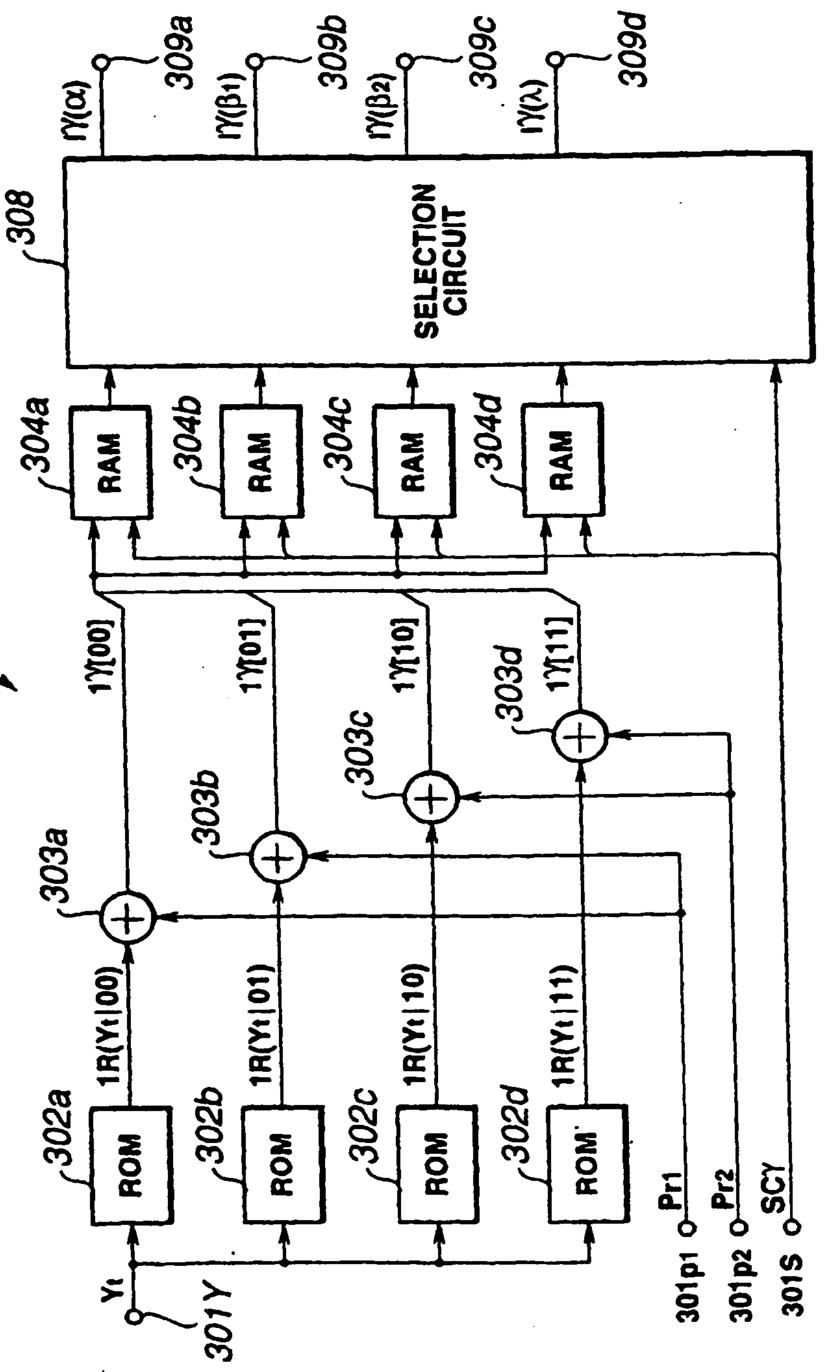

43
301Y
Yt
ROM
302a
302b
302c
302d
1R(Yt|00)
1R(Yt|01)
1R(Yt|10)
1R(Yt|11)
303a
303b
303c
303d
1Υ[00]
1Υ[01]
1Υ[10]
1Υ[11]
RAM
304a
304b
304c
304d
308
SELECTION CIRCUIT
1Υ(α)
1Υ(β1)
1Υ(β2)
1Υ(λ)
309a
309b
309c
309d
Pr1
Pr2
SCΥ
301p1
301p2
301S

FIG.8

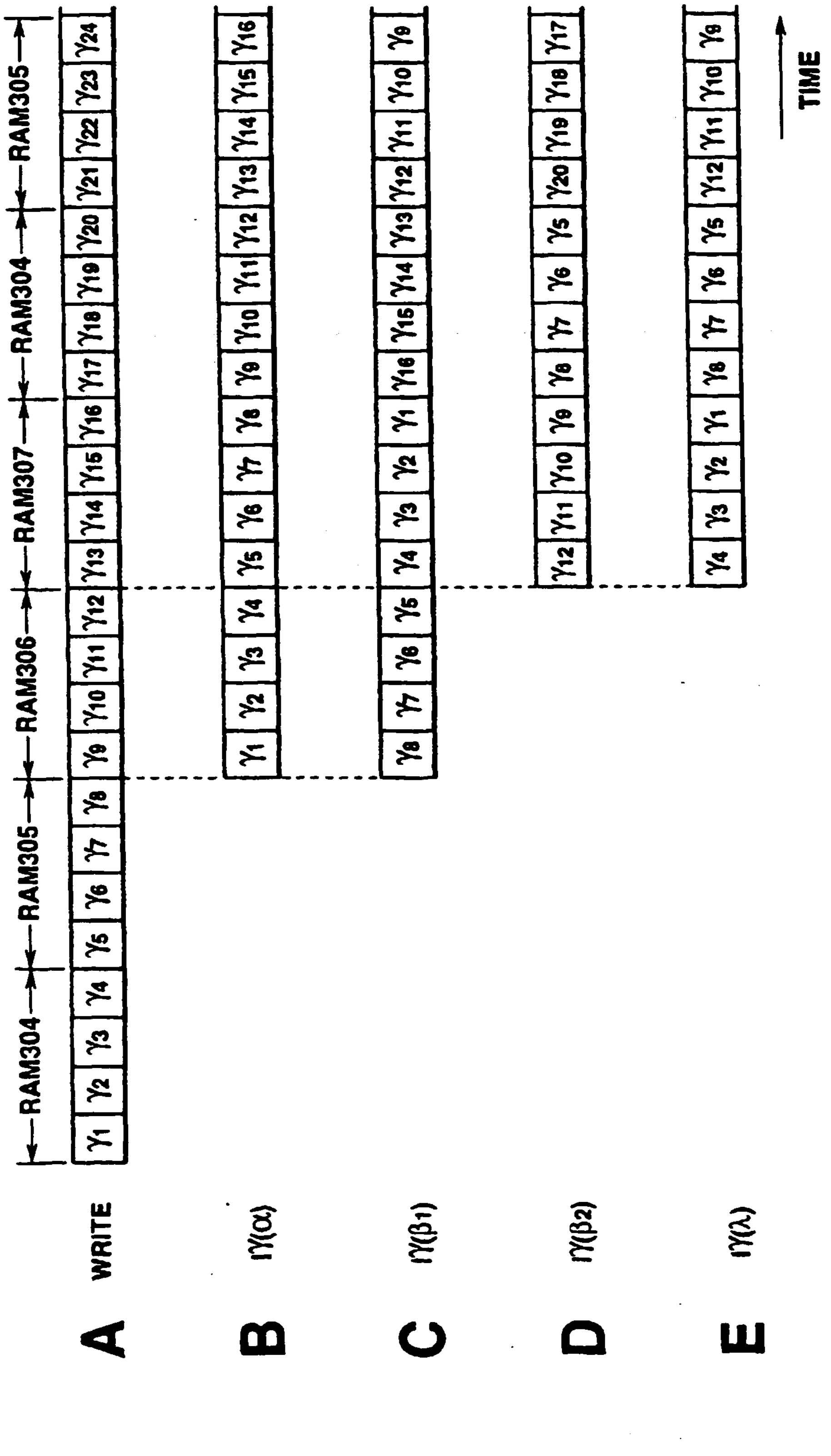
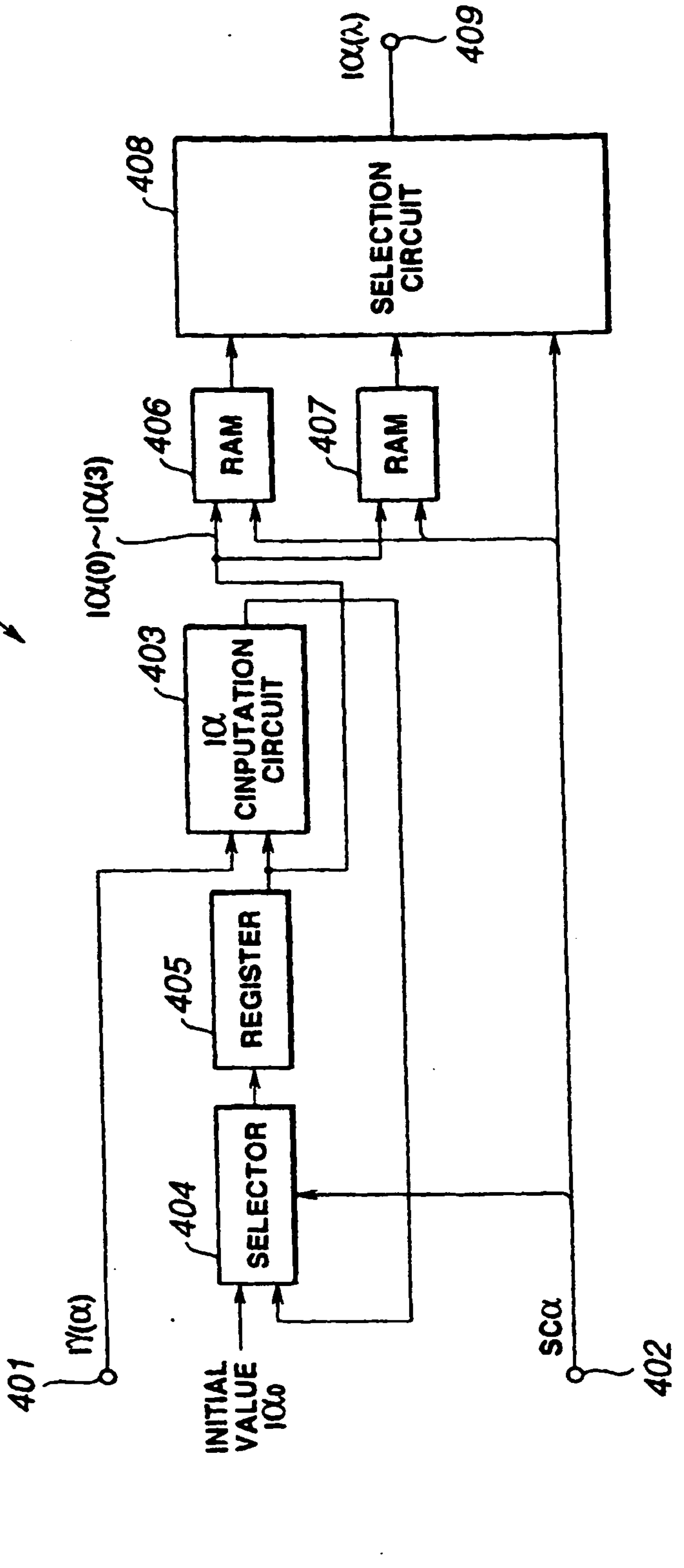

A WRITE
B IY(α)
C IY(β1)
D IY(β2)
E IY(λ)
RAM304
RAM305
RAM306
RAM307
RAM304
RAM305
Y1 Y2 Y3 Y4 Y5 Y6 Y7 Y8 Y9 Y10 Y11 Y12 Y13 Y14 Y15 Y16 Y17 Y18 Y19 Y20 Y21 Y22 Y23 Y24
Y1 Y2 Y3 Y4 Y5 Y6 Y7 Y8 Y9 Y10 Y11 Y12 Y13 Y14 Y15 Y16
Y8 Y7 Y6 Y5 Y4 Y3 Y2 Y1 Y16 Y15 Y14 Y13 Y12 Y11 Y10 Y9
Y12 Y11 Y10 Y9 Y8 Y7 Y6 Y5 Y20 Y19 Y18 Y17
Y4 Y3 Y2 Y1 Y8 Y7 Y6 Y5 Y12 Y11 Y10 Y9
TIME

FIG.9



44
401
Iγ(α)
INITIAL VALUE Iα0
404
SELECTOR
405
REGISTER
403
Iα CINPUTATION CIRCUIT
Iα(0)~Iα(3)
406
RAM
407
RAM
408
SELECTION CIRCUIT
Iα(λ)
409
SCα
402

FIG.10

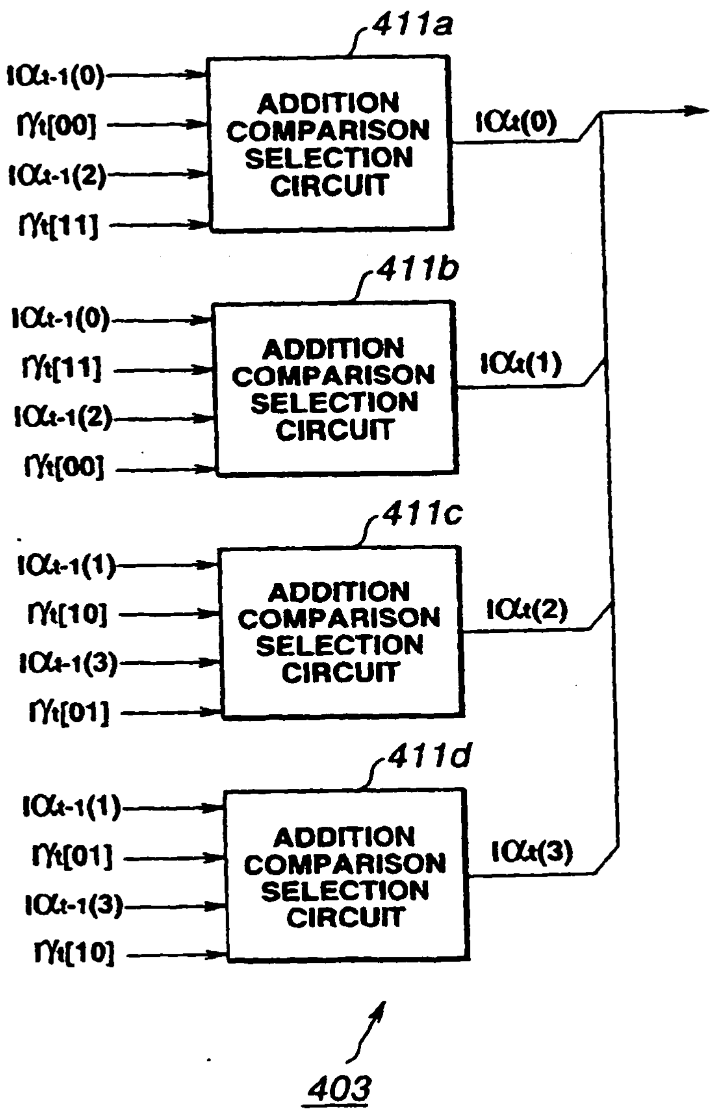

411a
Iαt-1(0)
Iγt[00]
Iαt-1(2)
Iγt[11]
ADDITION COMPARISON SELECTION CIRCUIT
Iαt(0)
411b
Iαt-1(0)
Iγt[11]
Iαt-1(2)
Iγt[00]
ADDITION COMPARISON SELECTION CIRCUIT
Iαt(1)
411c
Iαt-1(1)
Iγt[10]
Iαt-1(3)
Iγt[01]
ADDITION COMPARISON SELECTION CIRCUIT
Iαt(2)
411d
Iαt-1(1)
Iγt[01]
Iαt-1(3)
Iγt[10]
ADDITION COMPARISON SELECTION CIRCUIT
Iαt(3)
403

FIG.11

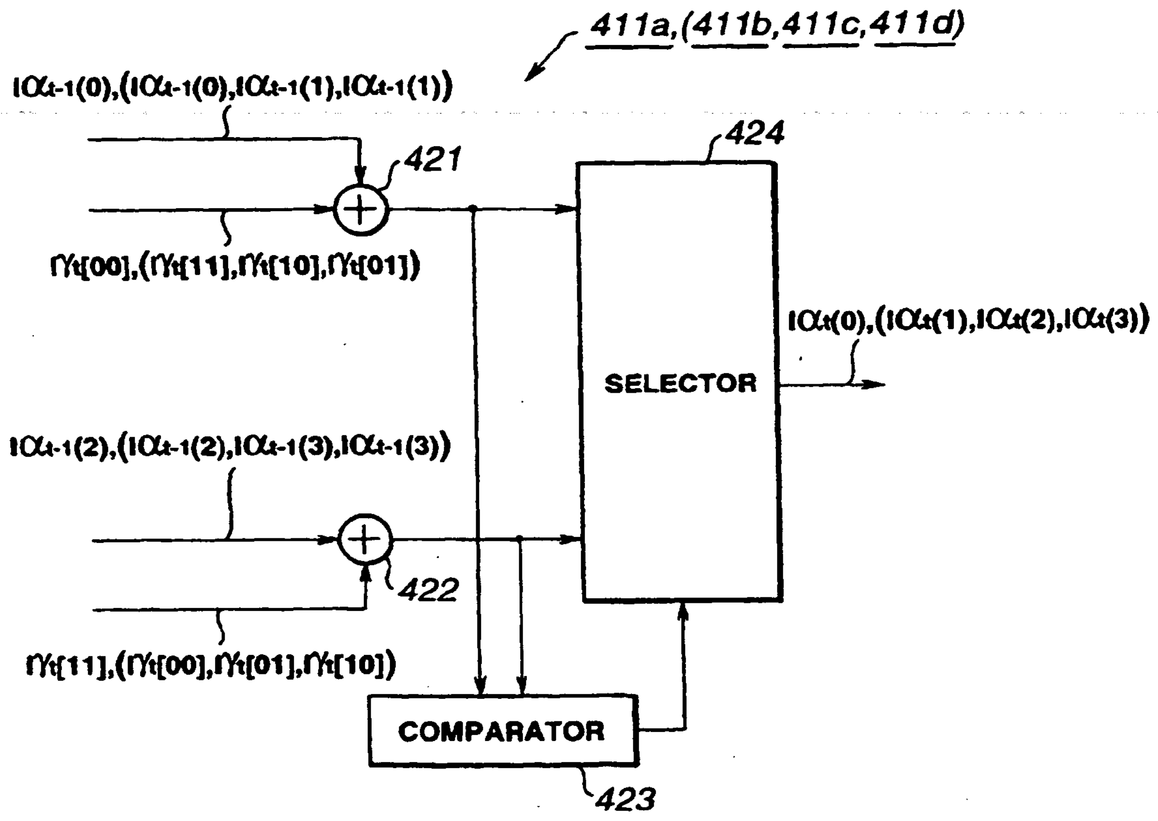

411a,(411b,411c,411d)
Iαt-1(0),(Iαt-1(0),Iαt-1(1),Iαt-1(1))
421
424
Iγt[00],(Iγt[11],Iγt[10],Iγt[01])
Iαt(0),(Iαt(1),Iαt(2),Iαt(3))
SELECTOR
Iαt-1(2),(Iαt-1(2),Iαt-1(3),Iαt-1(3))
422
Iγt[11],(Iγt[00],Iγt[01],Iγt[10])
COMPARATOR
423

FIG.12

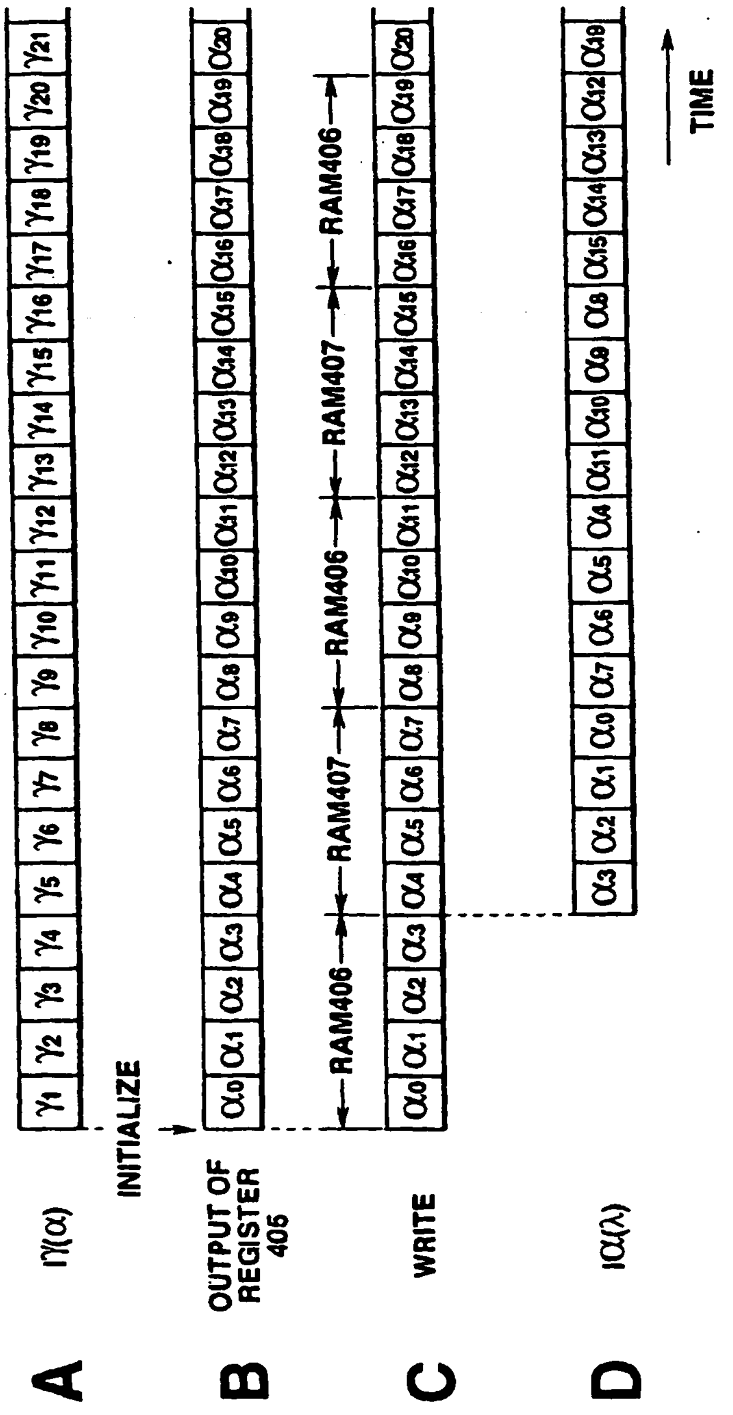

A
B
C
D
γ(α)
INITIALIZE
OUTPUT OF REGISTER 405
WRITE
α(λ)
RAM406
RAM407
RAM406
RAM407
RAM406
TIME
γ1 γ2 γ3 γ4 γ5 γ6 γ7 γ8 γ9 γ10 γ11 γ12 γ13 γ14 γ15 γ16 γ17 γ18 γ19 γ20 γ21
α0 α1 α2 α3 α4 α5 α6 α7 α8 α9 α10 α11 α12 α13 α14 α15 α16 α17 α18 α19 α20
α0 α1 α2 α3 α4 α5 α6 α7 α8 α9 α10 α11 α12 α13 α14 α15 α16 α17 α18 α19 α20
α3 α2 α1 α0 α7 α6 α5 α4 α11 α10 α9 α8 α15 α14 α13 α12 α19

FIG.13

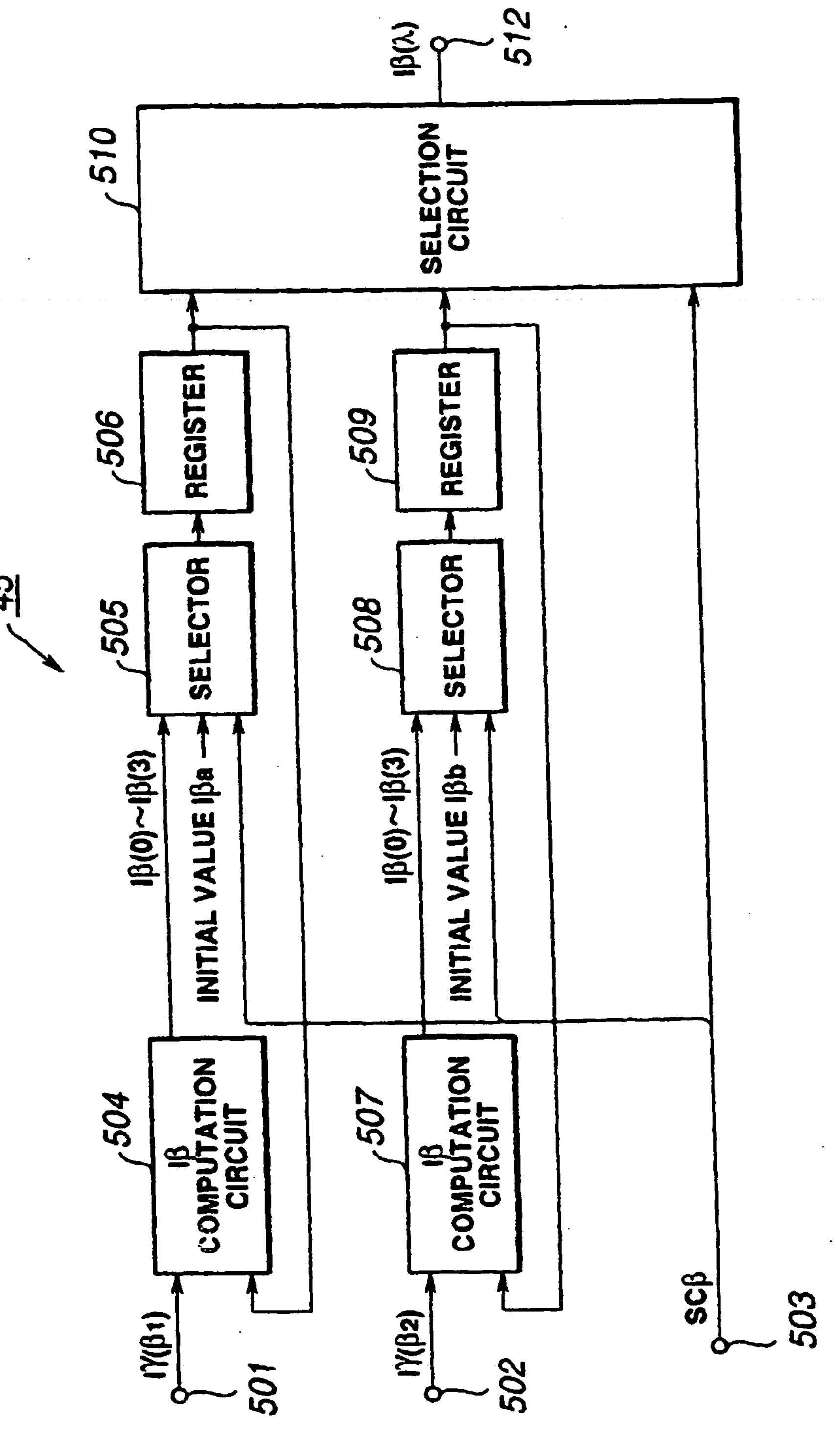

45
501
Iγ(β1)
502
Iγ(β2)
503
SCβ
504
Iβ COMPUTATION CIRCUIT
507
Iβ COMPUTATION CIRCUIT
Iβ(0)~Iβ(3)
INITIAL VALUE Iβa
Iβ(0)~Iβ(3)
INITIAL VALUE Iβb
505
SELECTOR
508
SELECTOR
506
REGISTER
509
REGISTER
510
SELECTION CIRCUIT
Iβ(λ)
512

FIG.14

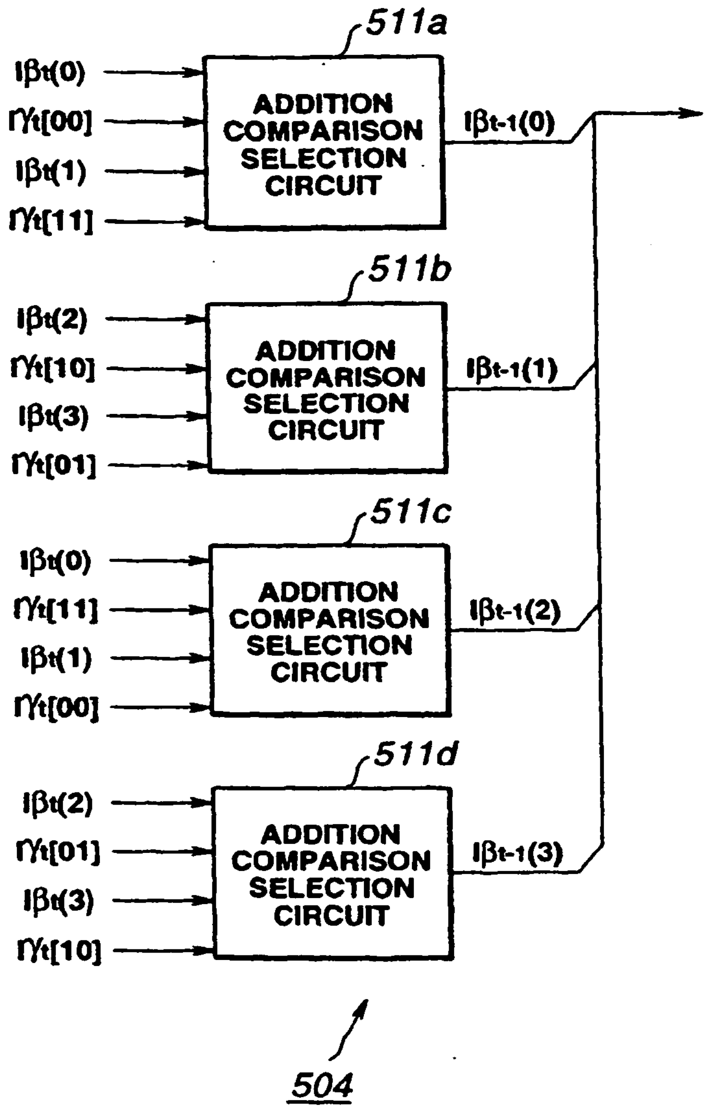

511a
Iβt(0)
Iγt[00]
Iβt(1)
Iγt[11]
ADDITION COMPARISON SELECTION CIRCUIT
Iβt-1(0)
511b
Iβt(2)
Iγt[10]
Iβt(3)
Iγt[01]
ADDITION COMPARISON SELECTION CIRCUIT
Iβt-1(1)
511c
Iβt(0)
Iγt[11]
Iβt(1)
Iγt[00]
ADDITION COMPARISON SELECTION CIRCUIT
Iβt-1(2)
511d
Iβt(2)
Iγt[01]
Iβt(3)
Iγt[10]
ADDITION COMPARISON SELECTION CIRCUIT
Iβt-1(3)
504

FIG.15

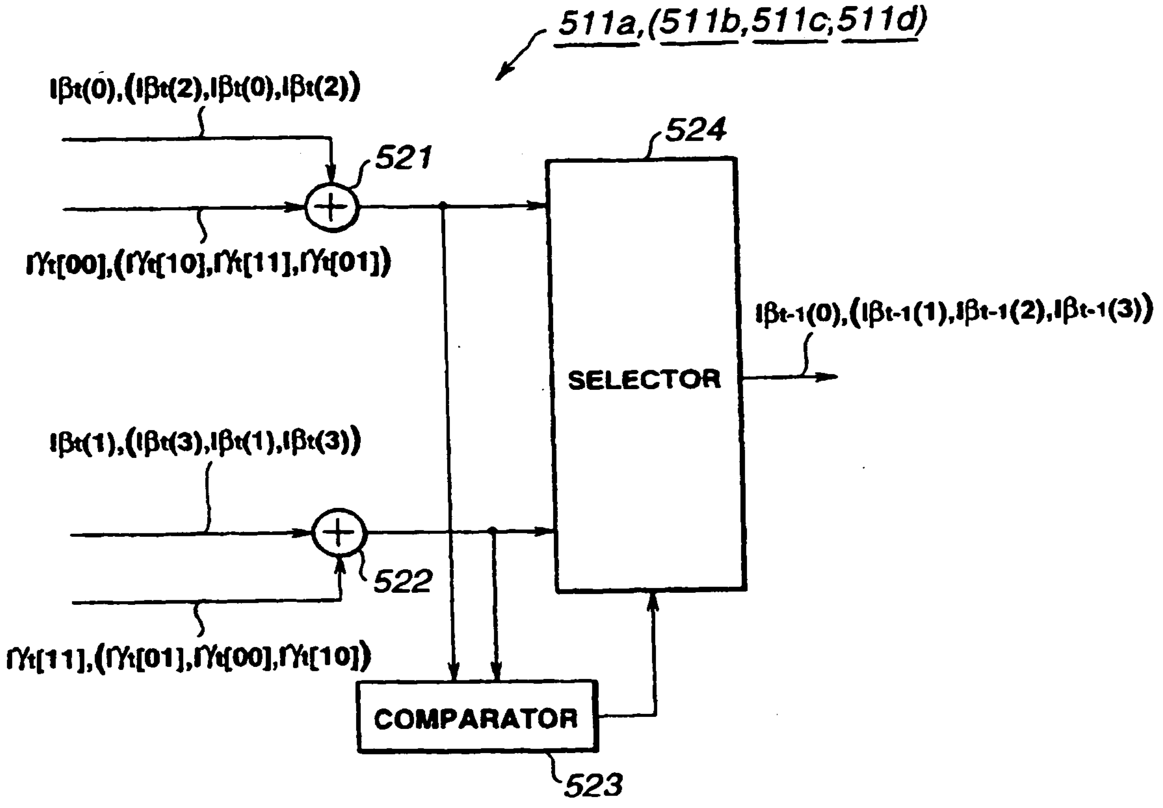

511a,(511b,511c,511d)
Iβt(0),(Iβt(2),Iβt(0),Iβt(2))
521
524
IYt[00],(IYt[10],IYt[11],IYt[01])
Iβt-1(0),(Iβt-1(1),Iβt-1(2),Iβt-1(3))
SELECTOR
Iβt(1),(Iβt(3),Iβt(1),Iβt(3))
522
IYt[11],(IYt[01],IYt[00],IYt[10])
COMPARATOR
523

FIG.16

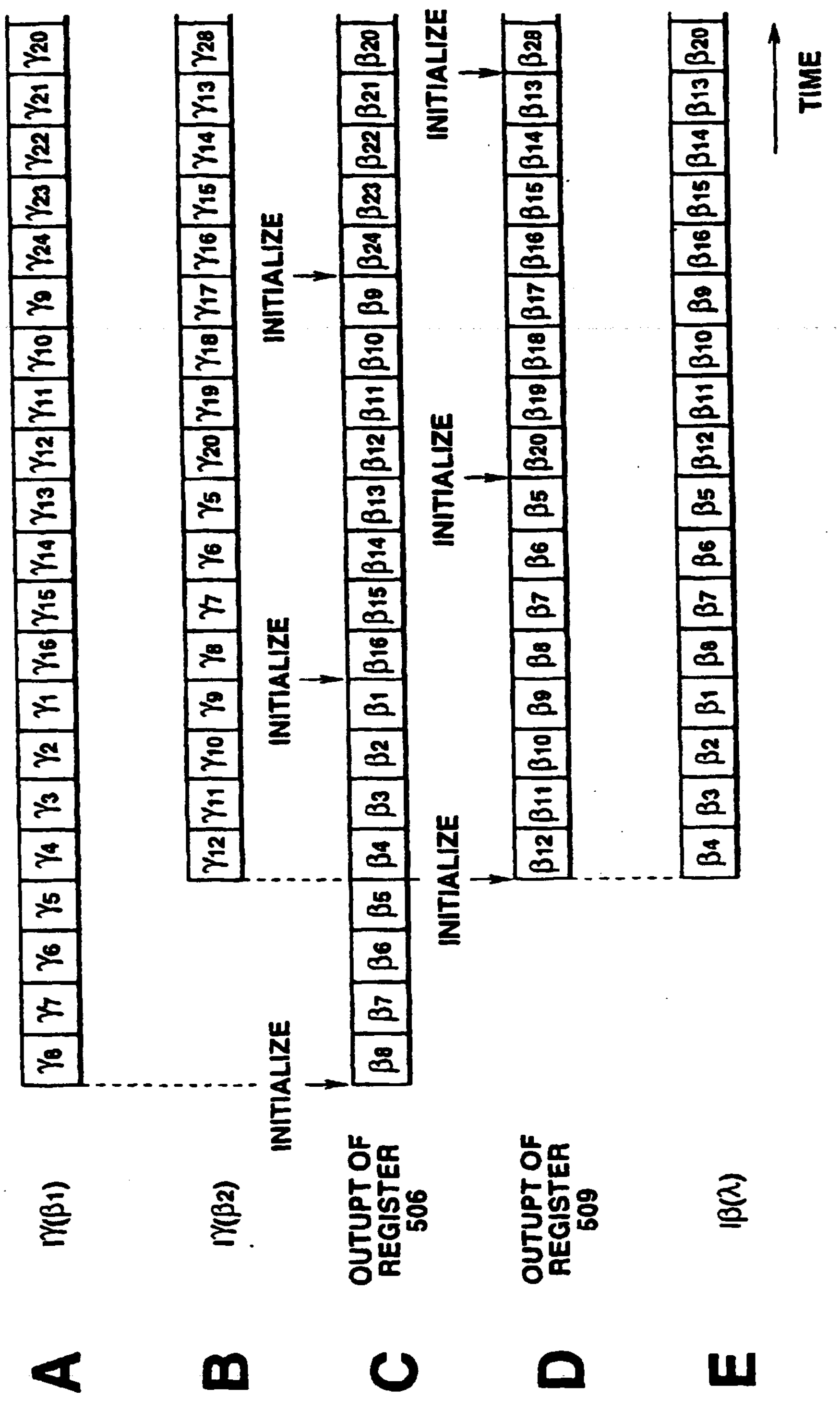

A
Iγ(β1)
B
Iγ(β2)
C
OUTUPT OF REGISTER 506
D
OUTUPT OF REGISTER 509
E
Iβ(λ)
INITIALIZE
INITIALIZE
INITIALIZE
INITIALIZE
INITIALIZE
INITIALIZE
TIME

FIG.17

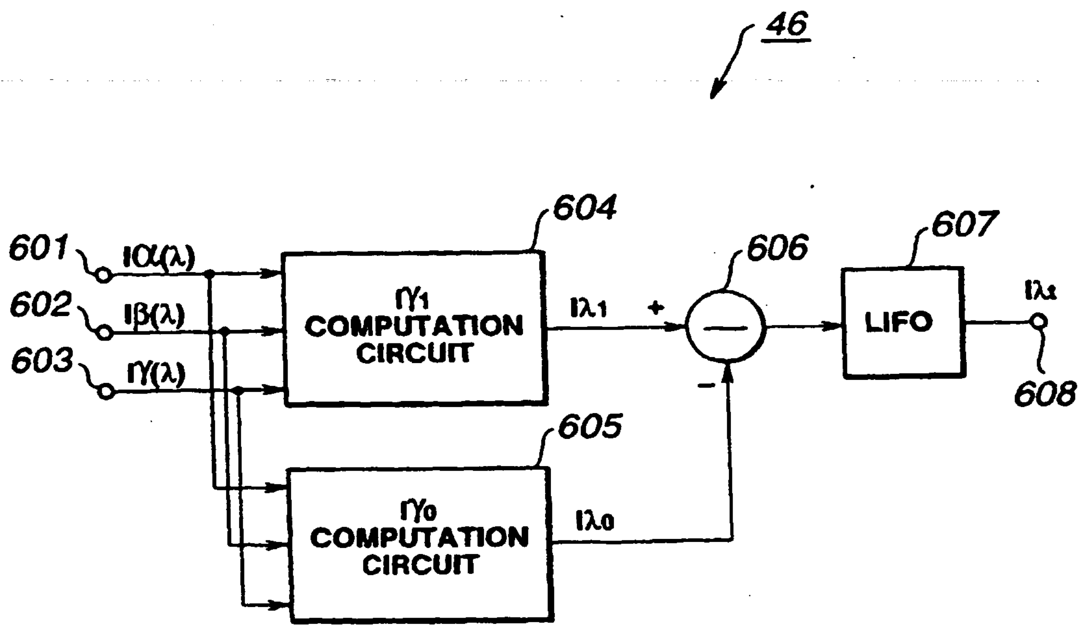

46
601
602
603
Iα(λ)
Iβ(λ)
Iγ(λ)
604
Iγ1 COMPUTATION CIRCUIT
Iλ1
+
606
−
607
LIFO
Iλt
608
605
Iγ0 COMPUTATION CIRCUIT
Iλ0

FIG.18

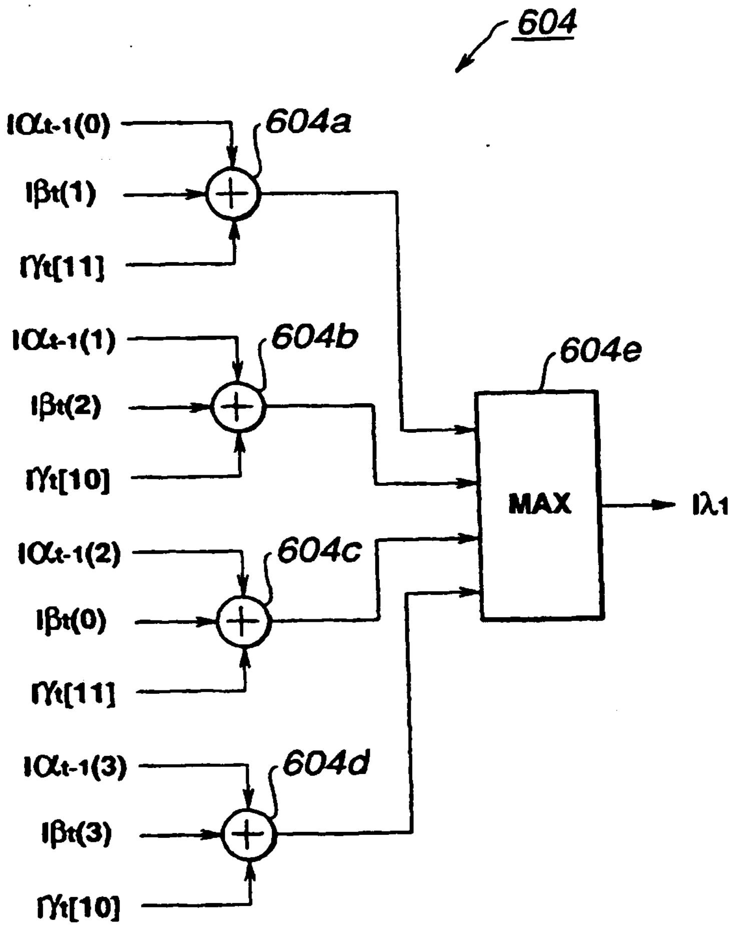

604
604a
604b
604c
604d
604e
Iαt-1(0)
Iβt(1)
Iγt[11]
Iαt-1(1)
Iβt(2)
Iγt[10]
Iαt-1(2)
Iβt(0)
Iγt[11]
Iαt-1(3)
Iβt(3)
Iγt[10]
MAX
Iλ1

FIG.19

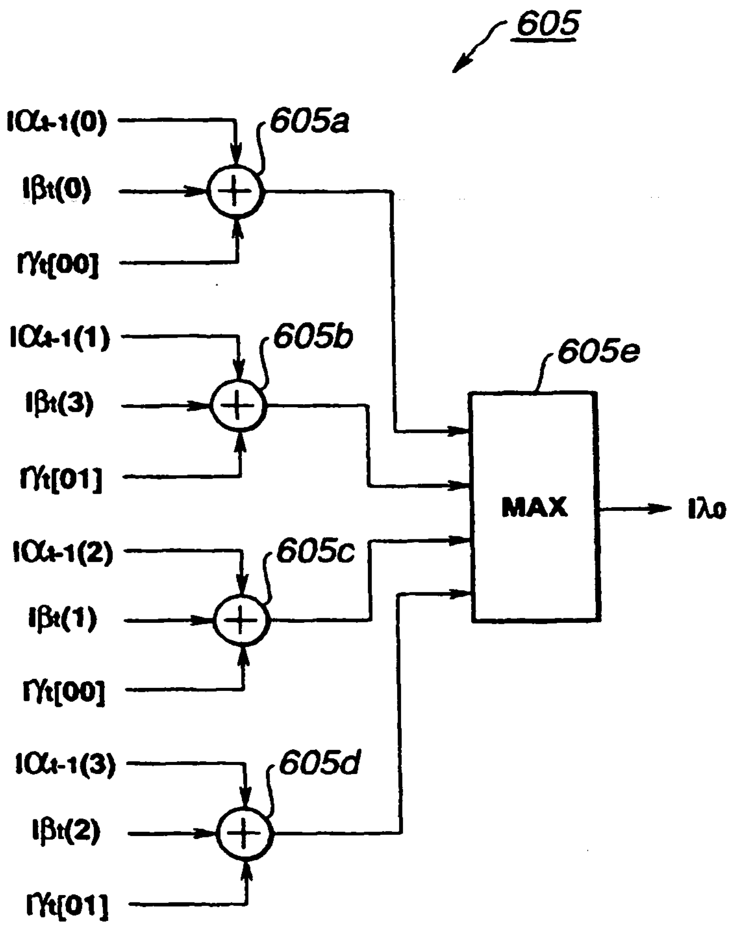

605
Iαt-1(0)
Iβt(0)
Iγt[00]
605a
Iαt-1(1)
Iβt(3)
Iγt[01]
605b
Iαt-1(2)
Iβt(1)
Iγt[00]
605c
Iαt-1(3)
Iβt(2)
Iγt[01]
605d
605e
MAX
Iλ0

FIG.20

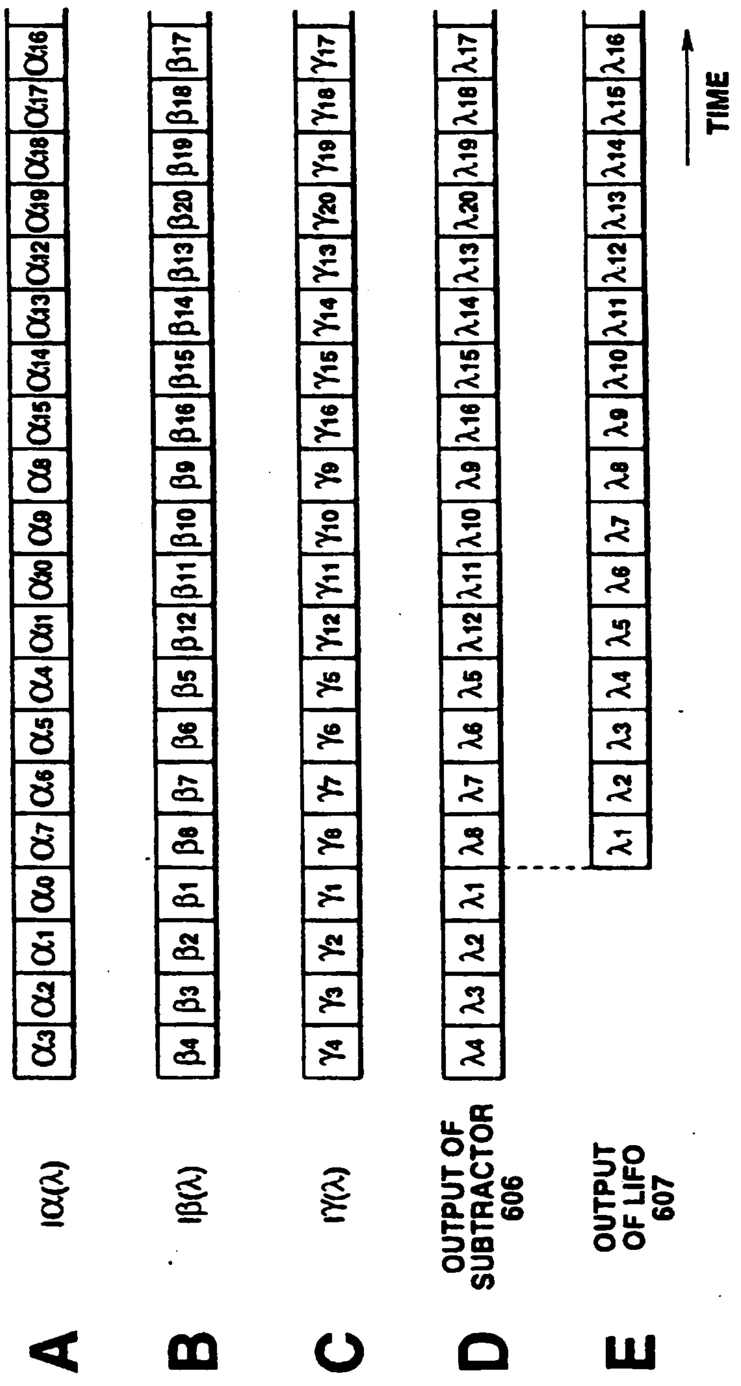

A
Iα(λ)
α3 α2 α1 α0 α7 α6 α5 α4 α11 α10 α9 α8 α15 α14 α13 α12 α19 α18 α17 α16
B
Iβ(λ)
β4 β3 β2 β1 β8 β7 β6 β5 β12 β11 β10 β9 β16 β15 β14 β13 β20 β19 β18 β17
C
Iγ(λ)
γ4 γ3 γ2 γ1 γ8 γ7 γ6 γ5 γ12 γ11 γ10 γ9 γ16 γ15 γ14 γ13 γ20 γ19 γ18 γ17
D
OUTPUT OF SUBTRACTOR 606
λ4 λ3 λ2 λ1 λ8 λ7 λ6 λ5 λ12 λ11 λ10 λ9 λ16 λ15 λ14 λ13 λ20 λ19 λ18 λ17
E
OUTPUT OF LIFO 607
λ1 λ2 λ3 λ4 λ5 λ6 λ7 λ8 λ9 λ10 λ11 λ12 λ13 λ14 λ15 λ16
TIME

FIG.21

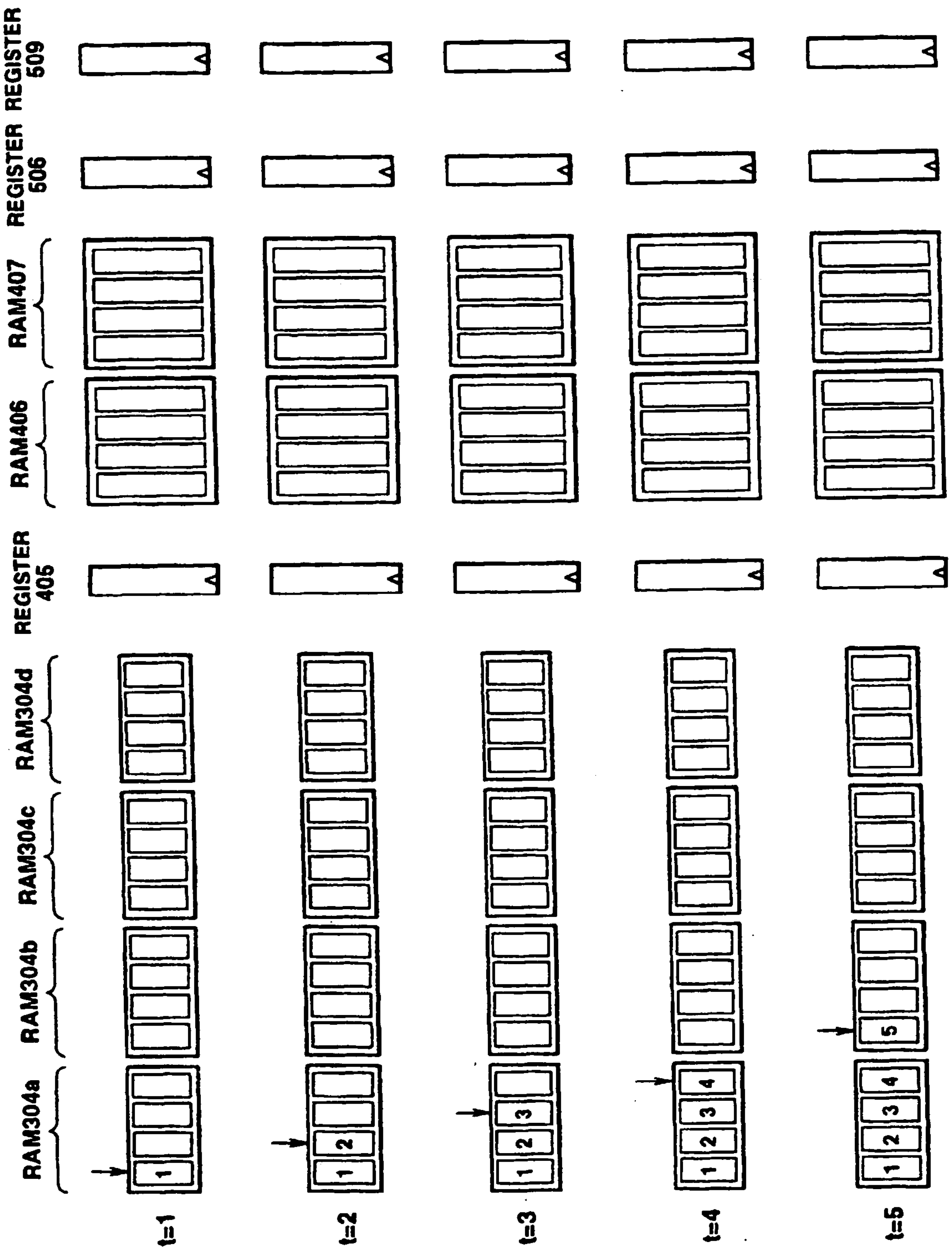

RAM304a
RAM304b
RAM304c
RAM304d
REGISTER 405
RAM406
RAM407
REGISTER 506
REGISTER 509
t=1
t=2
t=3
t=4
t=5
1
2
3
4
5

FIG.22A

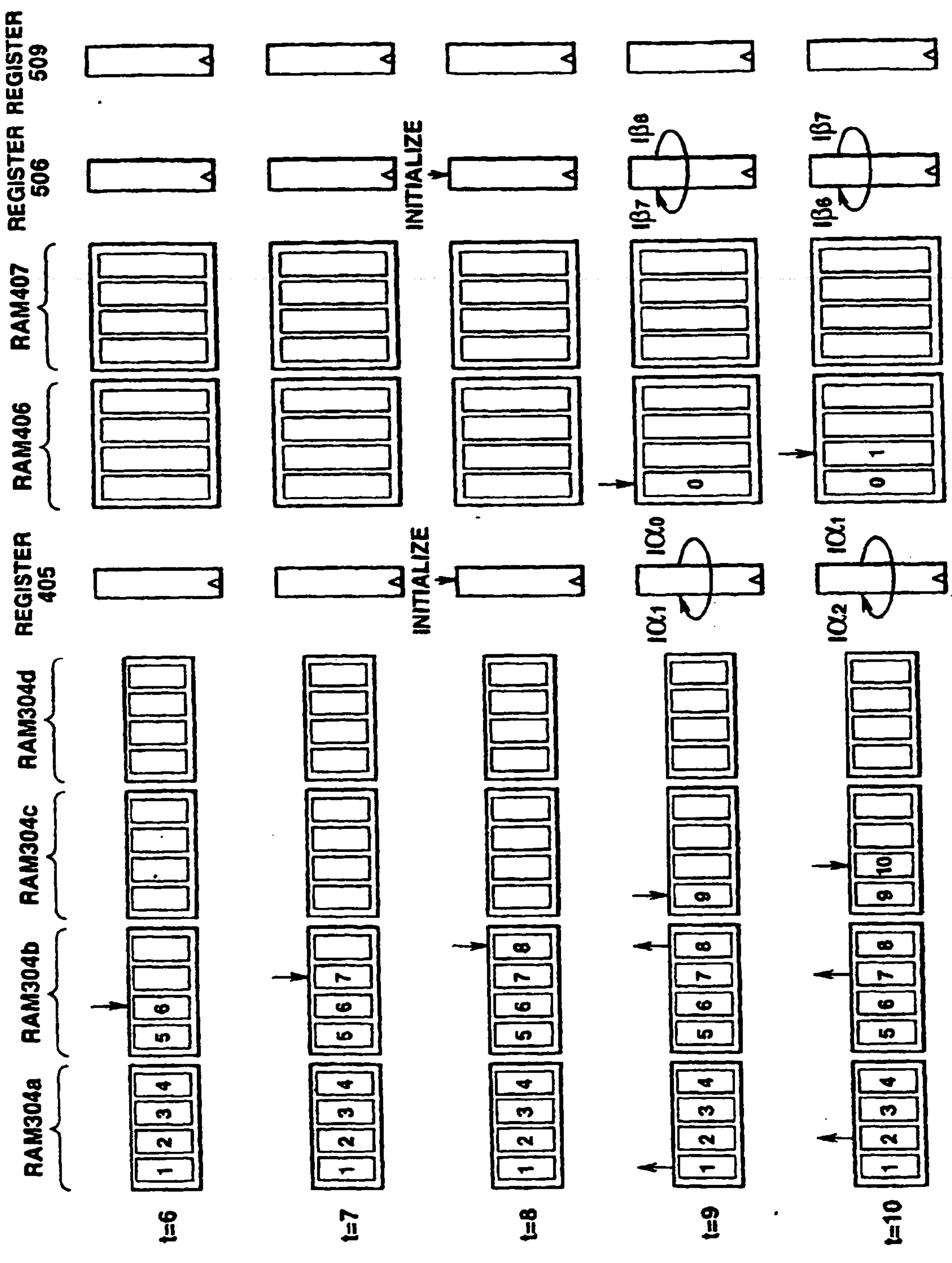
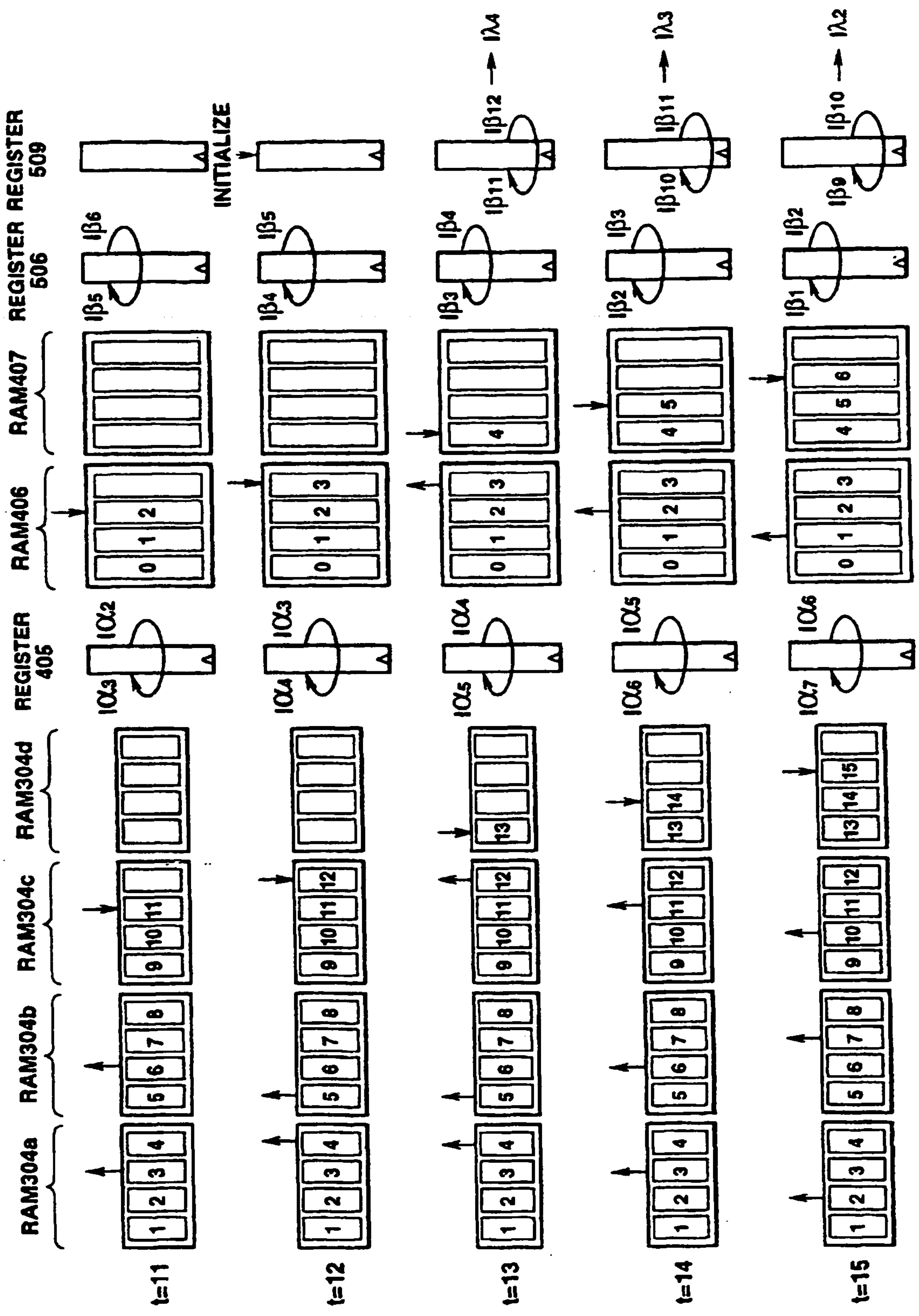

RAM304a
RAM304b
RAM304c
RAM304d
REGISTER 405
RAM406
RAM407
REGISTER 506
REGISTER 509
t=6
t=7
t=8
t=9
t=10
INITIALIZE
INITIALIZE
1 2 3 4
5 6 7 8
9 10
0 1
Iα0
Iα1
Iα2
Iβ6
Iβ7
Iβ8

FIG.22B



RAM304a
RAM304b
RAM304c
RAM304d
REGISTER 405
RAM406
RAM407
REGISTER 506
REGISTER 509
t=11
t=12
t=13
t=14
t=15
INITIALIZE
1α3
1α2
1α4
1α5
1α6
1α7
1β5
1β6
1β4
1β3
1β2
1β1
1β11
1β12
1β10
1β9
1λ4
1λ3
1λ2

FIG.22C

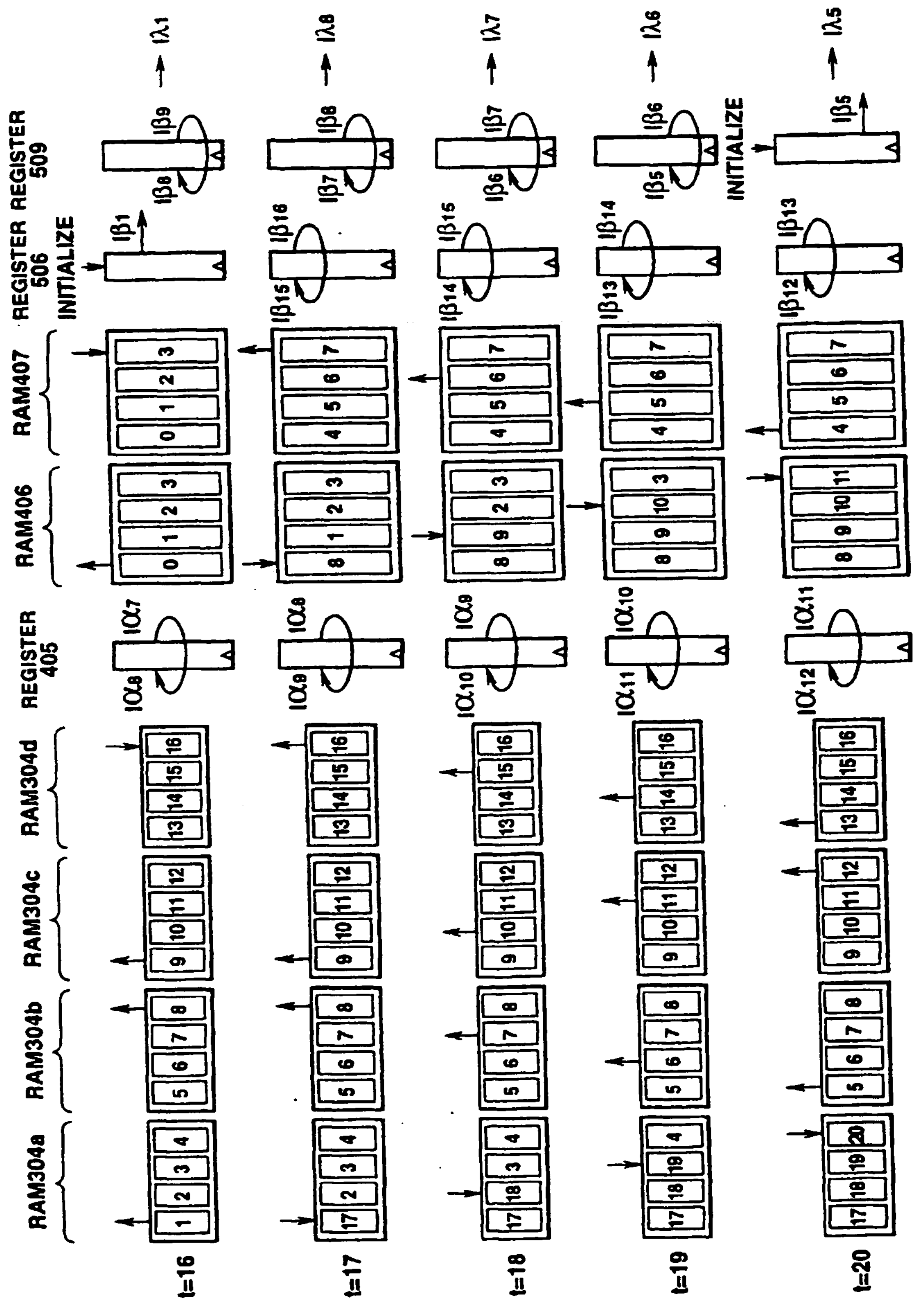

RAM304a
RAM304b
RAM304c
RAM304d
REGISTER 405
RAM406
RAM407
REGISTER 506
INITIALIZE
REGISTER 509
INITIALIZE
t=16
t=17
t=18
t=19
t=20
$I\alpha_7$
$I\alpha_8$
$I\alpha_9$
$I\alpha_{10}$
$I\alpha_{11}$
$I\alpha_{12}$
$I\beta_1$
$I\beta_{16}$
$I\beta_{15}$
$I\beta_{14}$
$I\beta_{13}$
$I\beta_{12}$
$I\beta_9$
$I\beta_8$
$I\beta_7$
$I\beta_6$
$I\beta_5$
$I\lambda_1$
$I\lambda_8$
$I\lambda_7$
$I\lambda_6$
$I\lambda_5$

FIG.22D

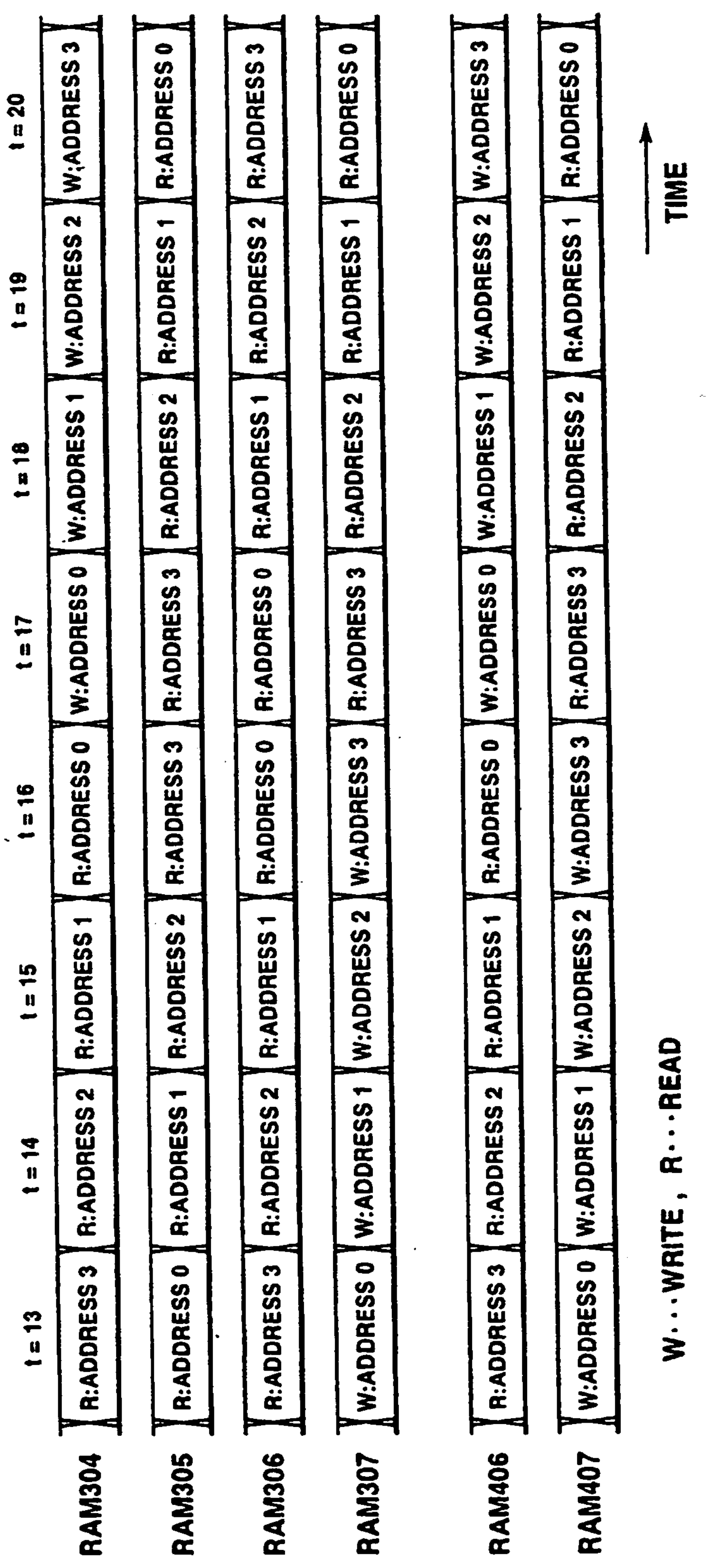
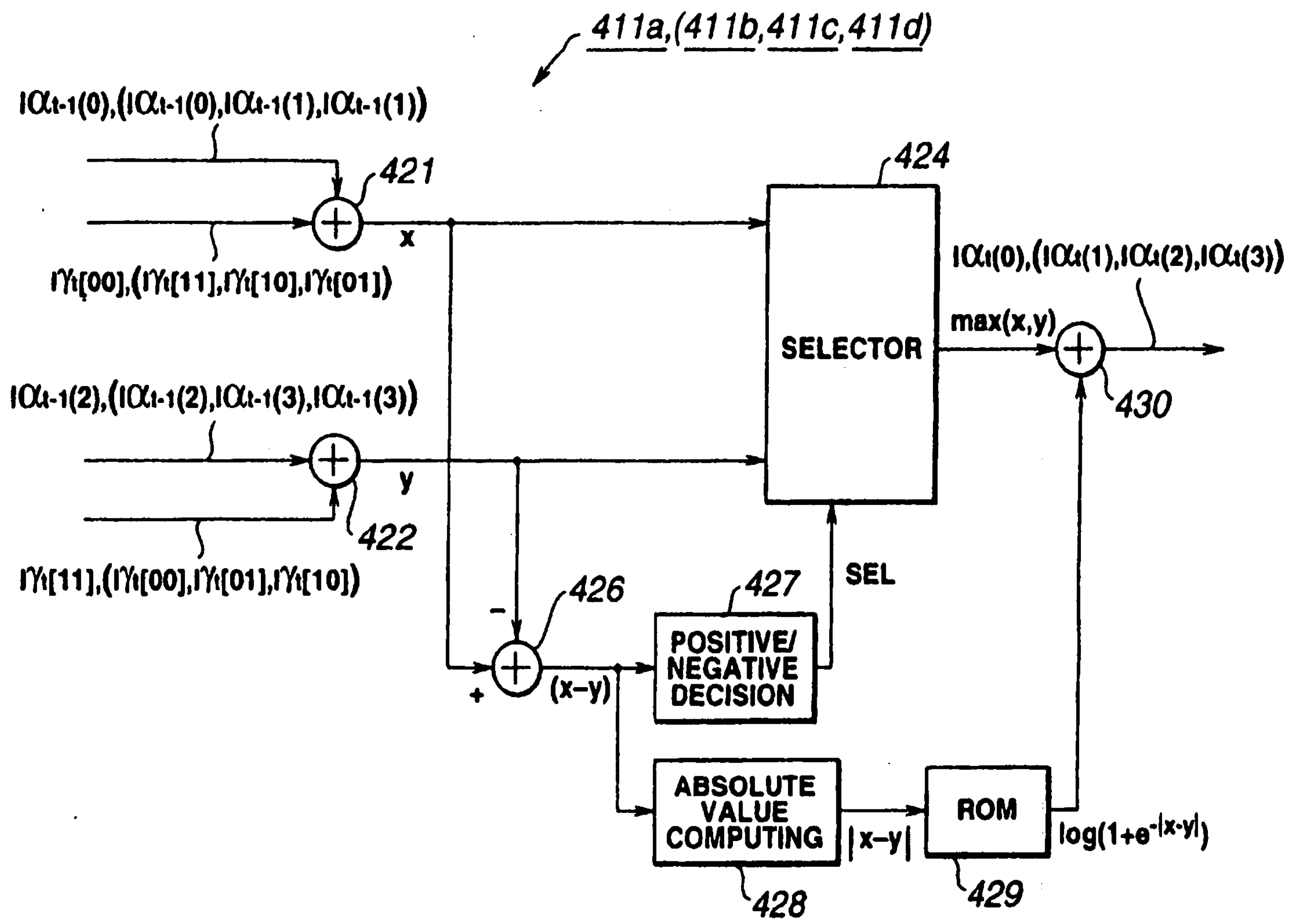

t = 13
t = 14
t = 15
t = 16
t = 17
t = 18
t = 19
t = 20
RAM304
R:ADDRESS 3
R:ADDRESS 2
R:ADDRESS 1
R:ADDRESS 0
W:ADDRESS 0
W:ADDRESS 1
W:ADDRESS 2
W:ADDRESS 3
RAM305
R:ADDRESS 0
R:ADDRESS 1
R:ADDRESS 2
R:ADDRESS 3
R:ADDRESS 3
R:ADDRESS 2
R:ADDRESS 1
R:ADDRESS 0
RAM306
R:ADDRESS 3
R:ADDRESS 2
R:ADDRESS 1
R:ADDRESS 0
R:ADDRESS 0
R:ADDRESS 1
R:ADDRESS 2
R:ADDRESS 3
RAM307
W:ADDRESS 0
W:ADDRESS 1
W:ADDRESS 2
W:ADDRESS 3
R:ADDRESS 3
R:ADDRESS 2
R:ADDRESS 1
R:ADDRESS 0
RAM406
R:ADDRESS 3
R:ADDRESS 2
R:ADDRESS 1
R:ADDRESS 0
W:ADDRESS 0
W:ADDRESS 1
W:ADDRESS 2
W:ADDRESS 3
RAM407
W:ADDRESS 0
W:ADDRESS 1
W:ADDRESS 2
W:ADDRESS 3
R:ADDRESS 3
R:ADDRESS 2
R:ADDRESS 1
R:ADDRESS 0
W···WRITE , R···READ
TIME

FIG.23



411a,(411b,411c,411d)
Iα_{t-1}(0),(Iα_{t-1}(0),Iα_{t-1}(1),Iα_{t-1}(1))
Iγ_t[00],(Iγ_t[11],Iγ_t[10],Iγ_t[01])
Iα_{t-1}(2),(Iα_{t-1}(2),Iα_{t-1}(3),Iα_{t-1}(3))
Iγ_t[11],(Iγ_t[00],Iγ_t[01],Iγ_t[10])
421
422
x
y
424
SELECTOR
Iα_t(0),(Iα_t(1),Iα_t(2),Iα_t(3))
max(x,y)
430
426
−
+
(x−y)
427
POSITIVE/ NEGATIVE DECISION
SEL
ABSOLUTE VALUE COMPUTING
428
|x−y|
ROM
429
log(1+e^{-|x-y|})

FIG.24

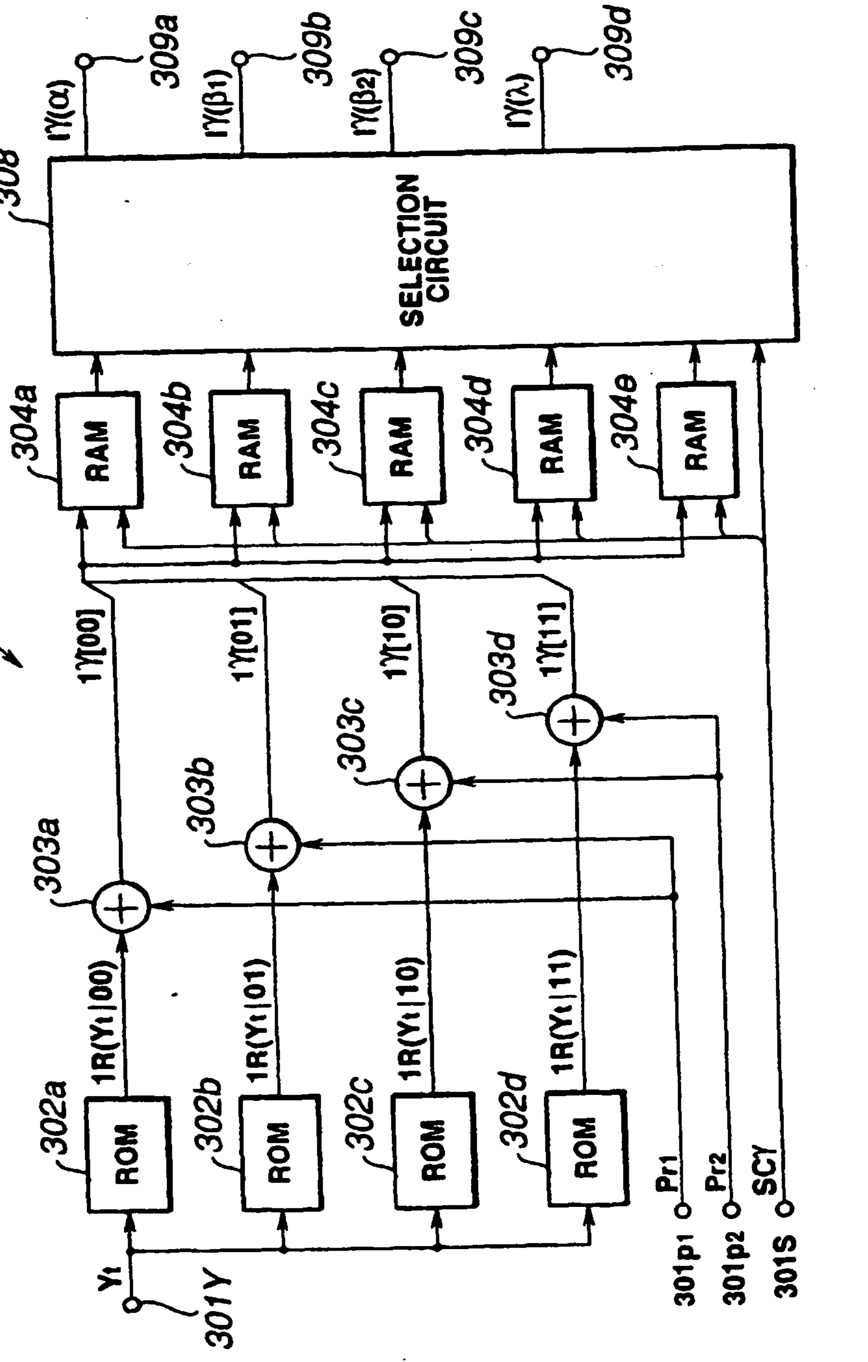

43
Yt
301Y
ROM
302a
302b
302c
302d
1R(Yt|00)
1R(Yt|01)
1R(Yt|10)
1R(Yt|11)
303a
303b
303c
303d
1Y[00]
1Y[01]
1Y[10]
1Y[11]
RAM
304a
304b
304c
304d
304e
SELECTION CIRCUIT
308
1Y(α)
1Y(β1)
1Y(β2)
1Y(λ)
309a
309b
309c
309d
Pr1
Pr2
SCY
301p1
301p2
301S

FIG.25

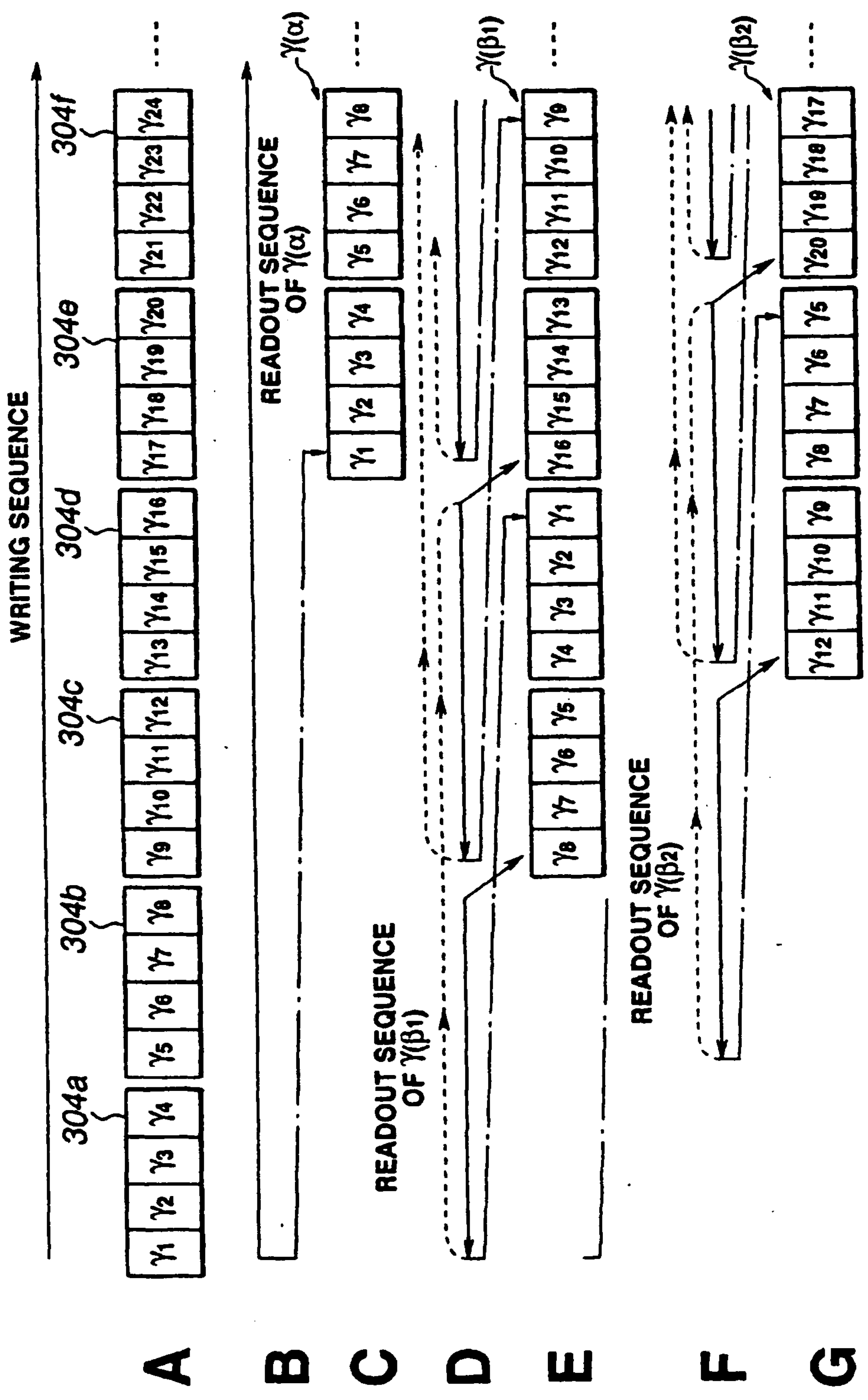

WRITING SEQUENCE
304a
304b
304c
304d
304e
304f
γ1 γ2 γ3 γ4 γ5 γ6 γ7 γ8 γ9 γ10 γ11 γ12 γ13 γ14 γ15 γ16 γ17 γ18 γ19 γ20 γ21 γ22 γ23 γ24
READOUT SEQUENCE OF γ(α)
γ(α)
γ1 γ2 γ3 γ4 γ5 γ6 γ7 γ8
READOUT SEQUENCE OF γ(β1)
γ(β1)
γ8 γ7 γ6 γ5 γ4 γ3 γ2 γ1 γ16 γ15 γ14 γ13 γ12 γ11 γ10 γ9
READOUT SEQUENCE OF γ(β2)
γ(β2)
γ12 γ11 γ10 γ9 γ8 γ7 γ6 γ5 γ20 γ19 γ18 γ17
A
B
C
D
E
F
G

FIG.26

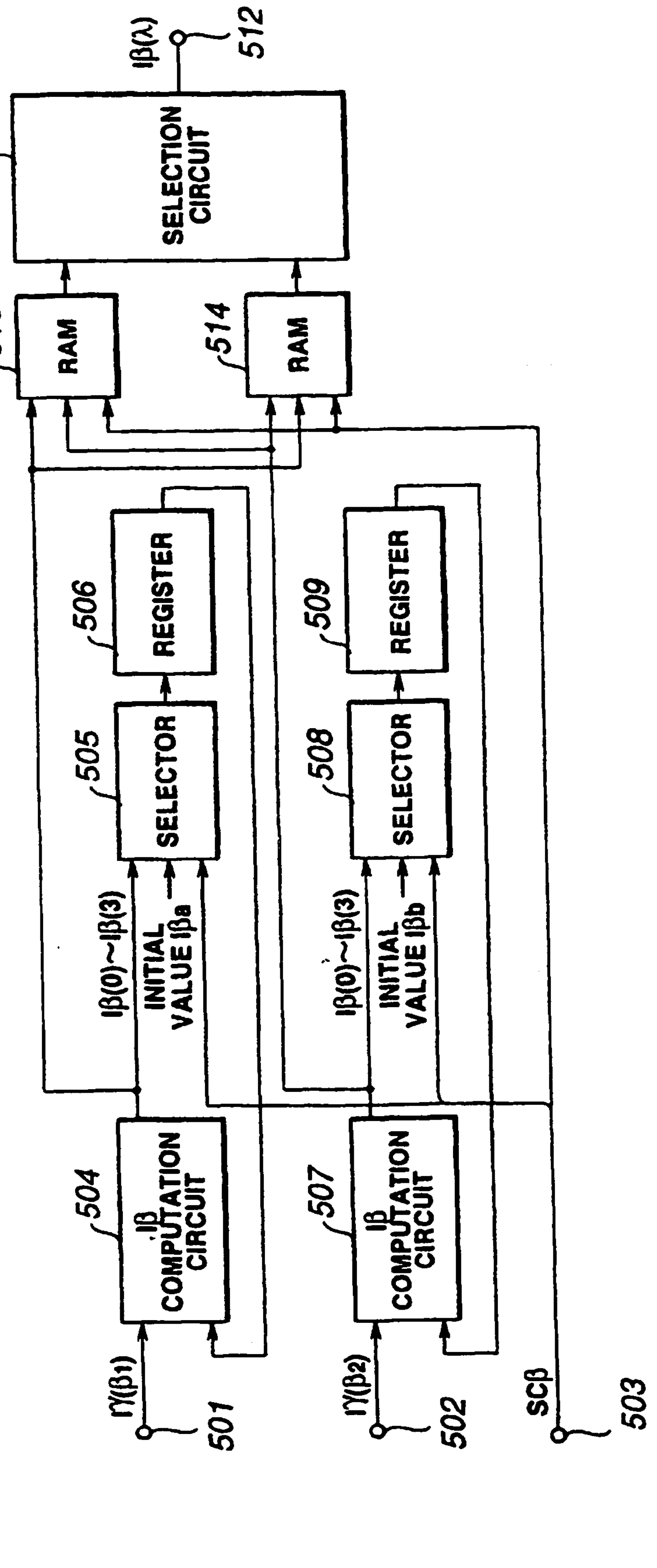

501
$I\gamma(\beta_1)$
502
$I\gamma(\beta_2)$
503
SCβ
504
Iβ COMPUTATION CIRCUIT
507
Iβ COMPUTATION CIRCUIT
505
SELECTOR
508
SELECTOR
$I\beta(0)\sim I\beta(3)$
INITIAL VALUE Iβa
INITIAL VALUE Iβb
506
REGISTER
509
REGISTER
513
RAM
514
RAM
510
SELECTION CIRCUIT
$I\beta(\lambda)$
512

FIG.27

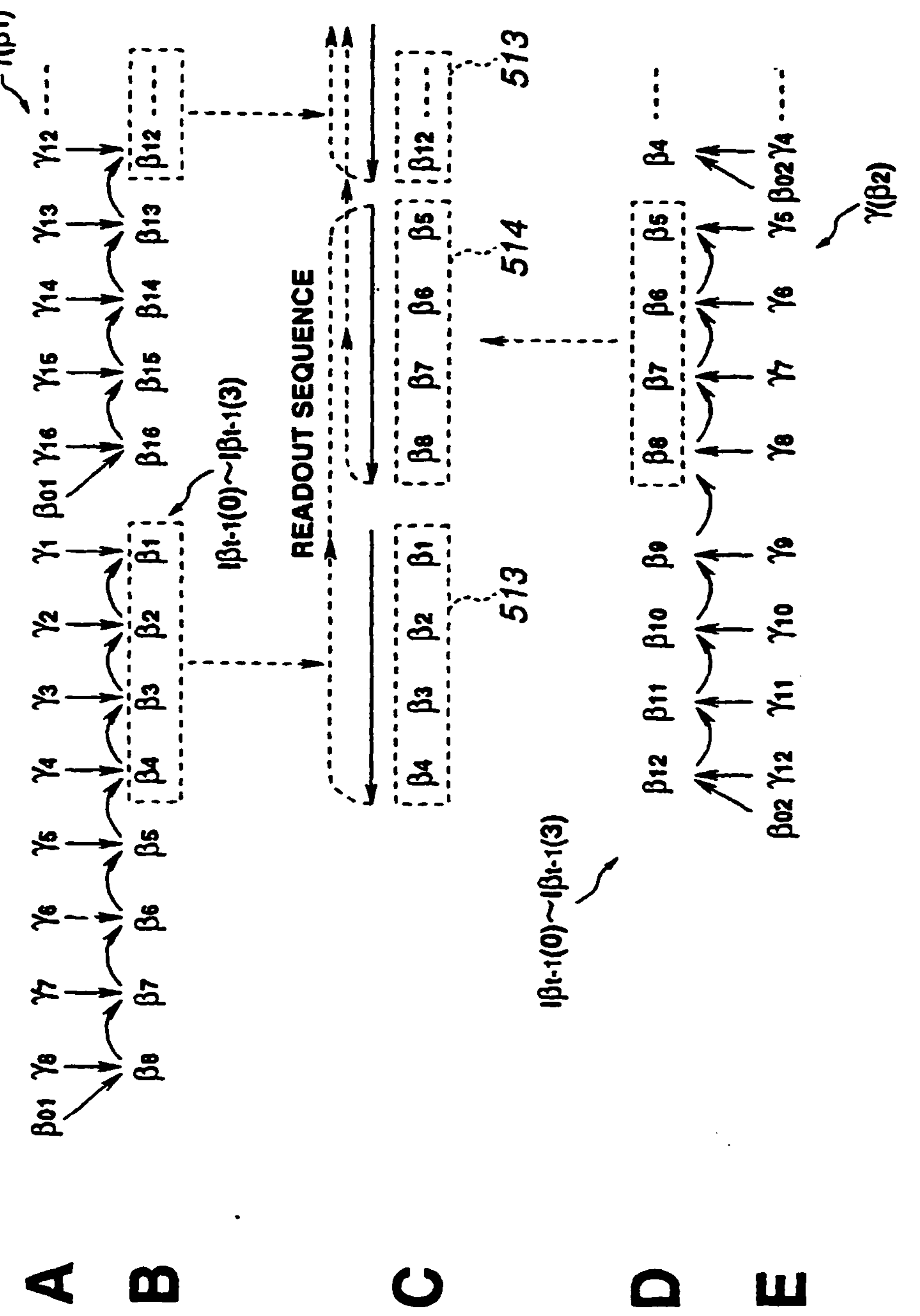

A
B
C
D
E
β01 γ8 γ7 γ6 γ5 γ4 γ3 γ2 γ1 β01 γ16 γ15 γ14 γ13 γ12 ----
γ(β1)
β8 β7 β6 β5 β4 β3 β2 β1 β16 β15 β14 β13 β12 ----
Iβt-1(0)~Iβt-1(3)
READOUT SEQUENCE
β4 β3 β2 β1
513
β8 β7 β6 β5
514
β12 ----
513
Iβt-1(0)~Iβt-1(3)
β12 β11 β10 β9 β8 β7 β6 β5 β4 ----
β02 γ12 γ11 γ10 γ9 γ8 γ7 γ6 γ5 β02 γ4 ----
γ(β2)

FIG.28

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP99/02553

A. CLASSIFICATION OF SUBJECT MATTER
Int.Cl^6 H03M13/12

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
Int.Cl^6 H03M13/12

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho (Y1, Y2)	1926-1996	Toroku Jitsuyo Shinan Koho (U)	1994-1999
Kokai Jitsuyo Shinan Koho (U)	1971-1999	Jitsuyo Shinan Toroku Koho (Y2)	1996-1999

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Chuu Yamamoto, Akira Fujiwara, Takuya Koumoto, Tadao Suu, "Saiki teki MAP Algorithm", Denshi Jouhou Tsuushin Gakkai Gijutsu Kenkyuu Houkoku, Vol. 96, No. 394 (IT96-44-46), (December, 1996), pp.1-6	1-13

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search 6 August, 1999 (06. 08. 99)	Date of mailing of the international search report 17 August, 1999 (17. 08. 99)
Name and mailing address of the ISA/ Japanese Patent Office Facsimile No.	Authorized officer Telephone No.

Form PCT/ISA/210 (second sheet) (July 1992)